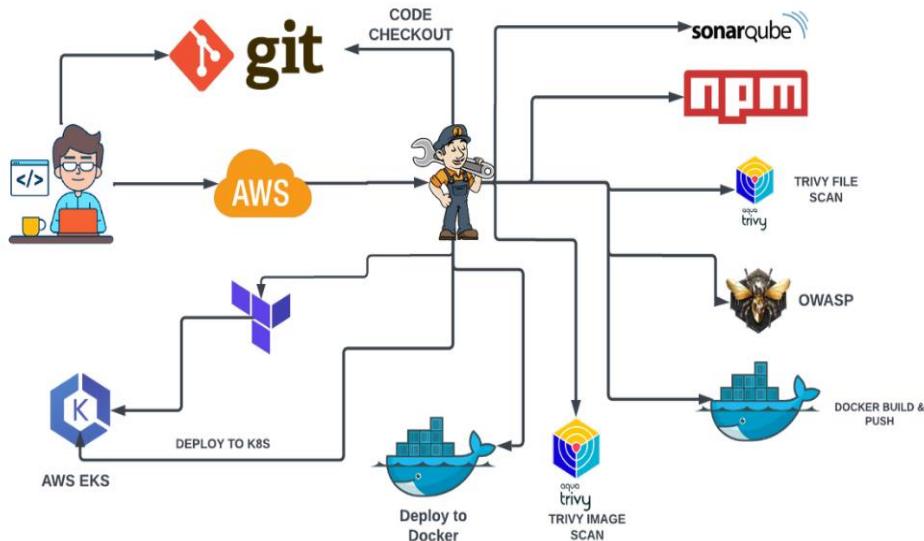


DevSecOps CI/CD Pipeline for an Uber Clone



GIT REPO: <https://github.com/vommidapuchinni/uber-clone.git>

Overview:

This document provides a detailed overview of the continuous integration and continuous deployment (CI/CD) pipeline implemented for the Uber clone application. It is intended for developers, DevOps engineers, and security analysts involved in maintaining and enhancing the pipeline.

Instructions to do this:

Step1: Launch an ubuntu instance (t2.2xlarge)

Step2: Create IAM role

Step3: Create S3 bucket

Step4: Installation of packages

Step5: Connect to Jenkins and SonarQube servers

Step6: Terraform plugin install and EKS provision

Step7: Plugins installation & setup (Java, Sonar, Nodejs, OWASP, Docker)

Step8: Configure in Global Tool Configuration

Step9: Configure Sonar Server in Manage Jenkins

Step10: Pipeline up to Docker

Step11: Kubernetes Deployment

Step12: Accessing our application

Step13: Destruction

Prerequisites:

1. Version Control System (VCS):

- Use a VCS like Git (GitHub) to manage your source code.
Ensure proper branching strategies are in place (e.g., feature branches, develop, master).

2. Containerization:

- Docker containerization tool to package your application and its dependencies into containers, which ensures consistency across different environments.

3. Infrastructure as Code (IaC):

- Tools like Terraform to define and manage your infrastructure in a version-controlled and repeatable manner.

4. CI/CD Tool:

- Choose a CI/CD tool like Jenkins, to automate the build, test, and deployment phases.

5. Security Tools Integration:

- Select security tools for static code analysis (e.g., SonarQube), image scanner (Trivy), OWASP dependency check.

Definitions:

1. **GitHub** is a web-based platform for Git repositories, enabling version control and collaborative software development
2. **Git** is a distributed version control system for tracking changes in source code during software development.
3. **AWS (Amazon Web Services)** is a comprehensive cloud computing platform offering on-demand computing resources and services.
4. **EC2 (Elastic Compute Cloud)** is a web service that provides resizable compute capacity in the cloud, designed to make web-scale computing easier for developers.
5. **Terraform** is an open-source infrastructure as code (IaC) tool used for building, changing, and versioning infrastructure safely and efficiently.

6. AWS EKS (Amazon Elastic Kubernetes Service) is a managed Kubernetes service provided by AWS for deploying, managing, and scaling containerized applications using Kubernetes on AWS cloud infrastructure.

7. Docker is a platform that enables developers to package, deploy, and run applications and their dependencies in lightweight, portable containers.

8. Trivy is an open-source vulnerability scanner for containers and other artifacts, used to detect security vulnerabilities in software dependencies.

Trivy supports both file scanning and image scanning to identify security vulnerabilities:

- ❖ File Scan: Trivy can scan individual files or directories to detect vulnerabilities in software packages and dependencies present within those files.
- ❖ Image Scan: Trivy can also scan Docker images to identify vulnerabilities in operating system packages, application dependencies, and other components within the container image layers.

9. OWASP Dependency-Check is an open-source tool that identifies known vulnerabilities in project dependencies, helping developers manage and mitigate security risks.

10. npm (Node Package Manager) is a package manager for JavaScript that helps developers discover, share, and reuse code, managing dependencies and scripts for Node.js projects.

11. SonarQube is an open-source platform for continuous inspection of code quality and security, providing detailed reports and analytics to improve software development practices.

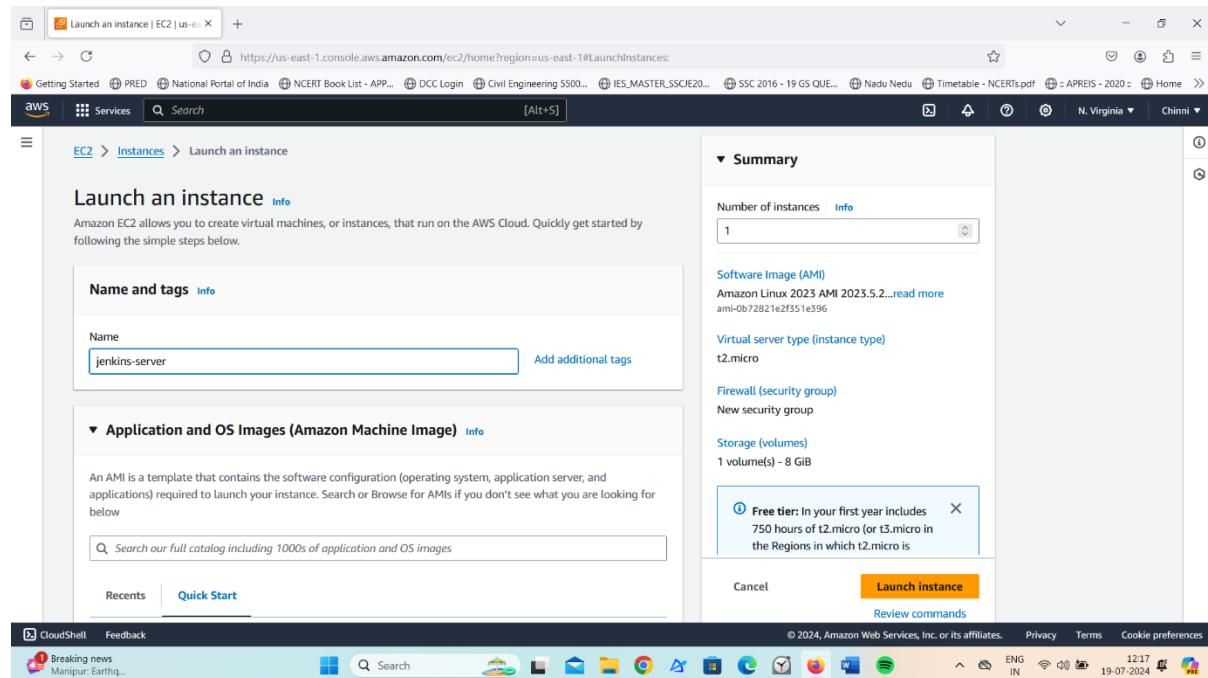
12. IAM role (Identity and Access Management role) defines permissions for AWS entities without the need for credentials, enhancing security and access control in cloud environments.

13. S3 (Simple Storage Service) is an object storage service offered by AWS, providing scalable storage for data, files, and backups accessible via the internet.

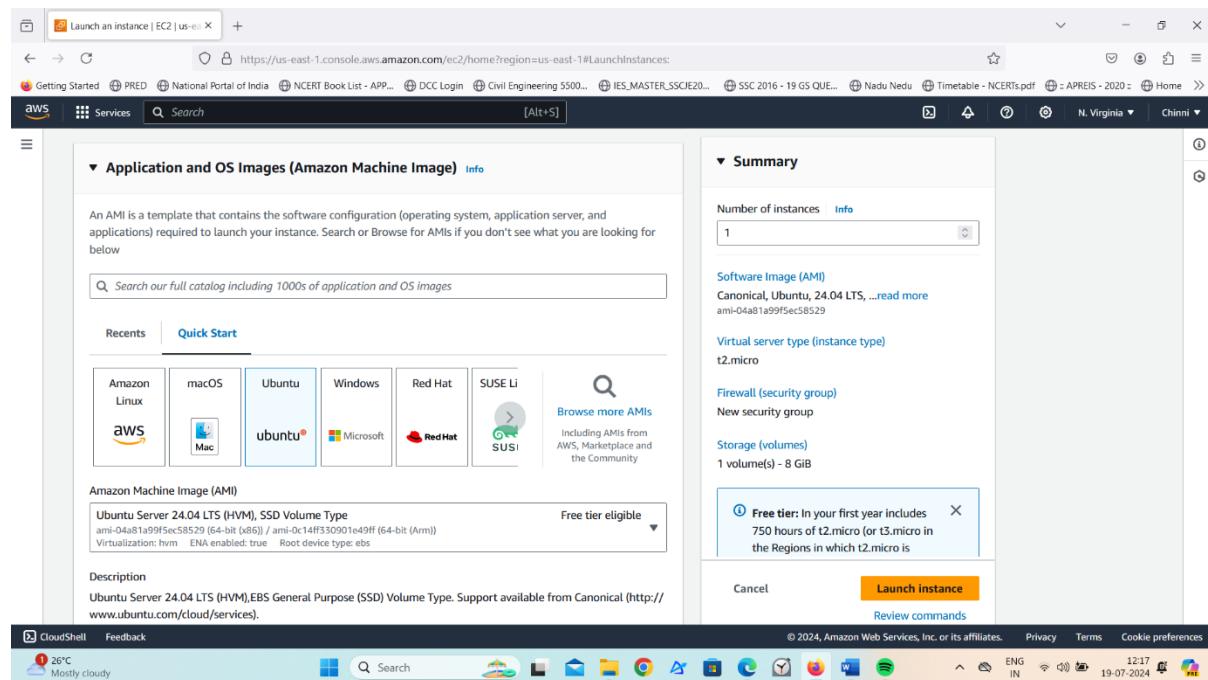
14. Jenkins is an open-source automation server used for continuous integration and continuous delivery (CI/CD), facilitating software development processes through automation of build, test, and deployment tasks.

Step1: Launch an ubuntu instance (t2.2xlarge)

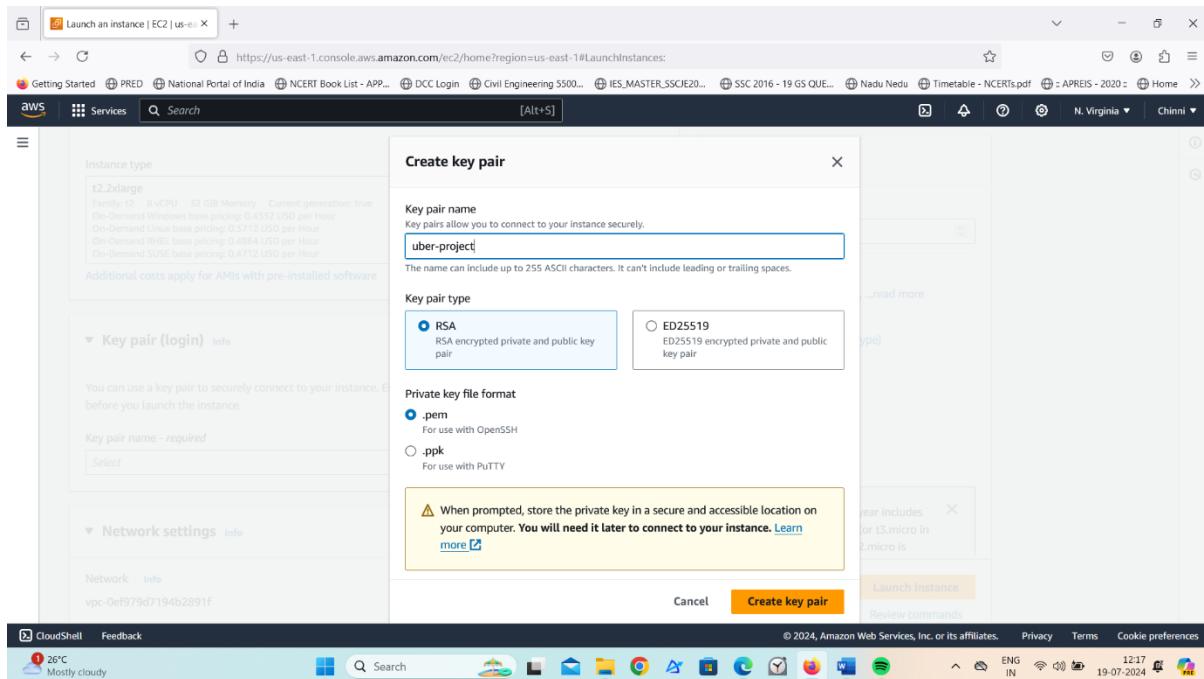
- Sign in to AWS Console: Log in to your AWS Management Console.
- Navigate to EC2 Dashboard.
- Launch Instance: Click on the “Launch Instance” button.



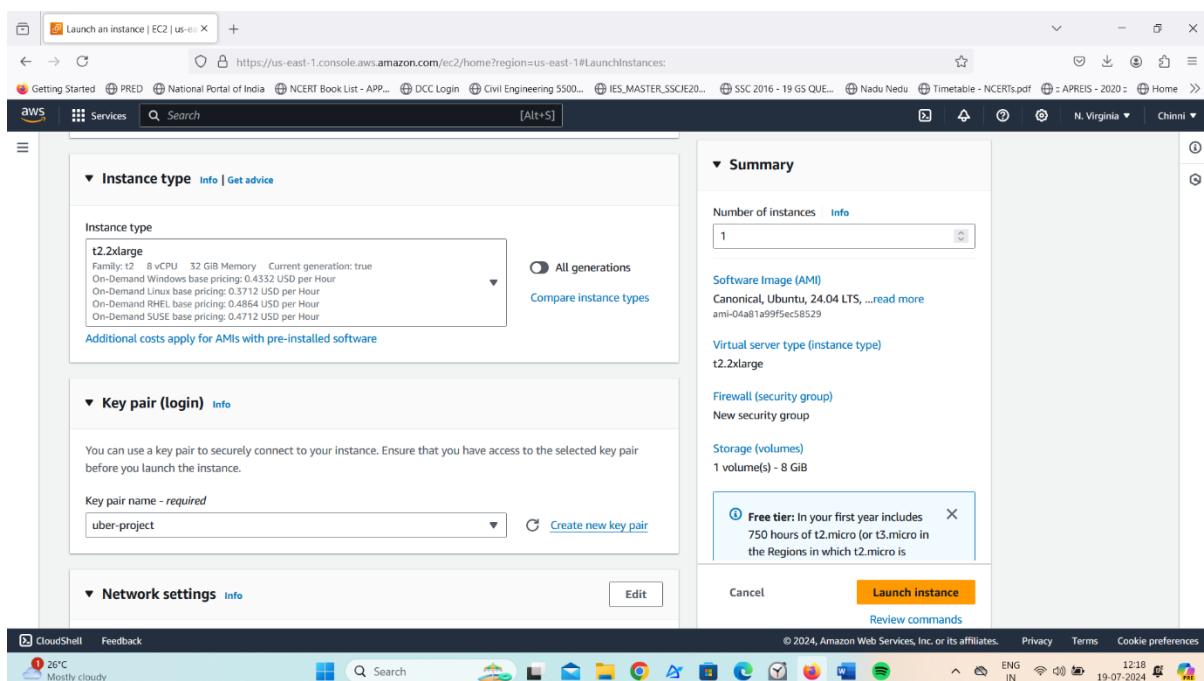
- Choose an Amazon Machine Image (AMI): Select an appropriate AMI for your instance. For example, you can choose Ubuntu image.



- Create new key pair.



- Choose an Instance Type: In the “Choose Instance Type” step, select t2.2xlarge as your instance type. Proceed by clicking “Next: Configure Instance Details.”



- Configure additional settings like network, subnets, security groups and allow inbound rule as all traffic (0.0.0.0/0)

[Launch an instance | EC2 | us-east-1](#)

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

Inbound Security Group Rules

- Security group rule 1 (TCP, 22, 0.0.0.0)
 - Type: ssh, Protocol: TCP, Port range: 22
 - Source type: Anywhere, Description: e.g. SSH for admin desktop
- Security group rule 2 (All, All, 0.0.0.0)
 - Type: All traffic, Protocol: All, Port range: All
 - Source type: Custom, Description: e.g. SSH for admin desktop

Rules with source of 0.0.0.0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04 LTS, ...read more
ami-04a81a9f5ec58529

Virtual server type (instance type): t2.2xlarge

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is available)

Cancel Launch instance Review commands

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 12:18 19-07-2024

[Instances | EC2 | us-east-1](#)

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

Instances (1) info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pu
jenkins-server	i-082e4110e988f7b6	Running	t2.2xlarge	Initializing		us-east-1a	ec2

Select an instance

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 12:19 19-07-2024

[Instance details | EC2 | us-east-1](#)

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetail\$instanceId=i-082e4110e988f7b6

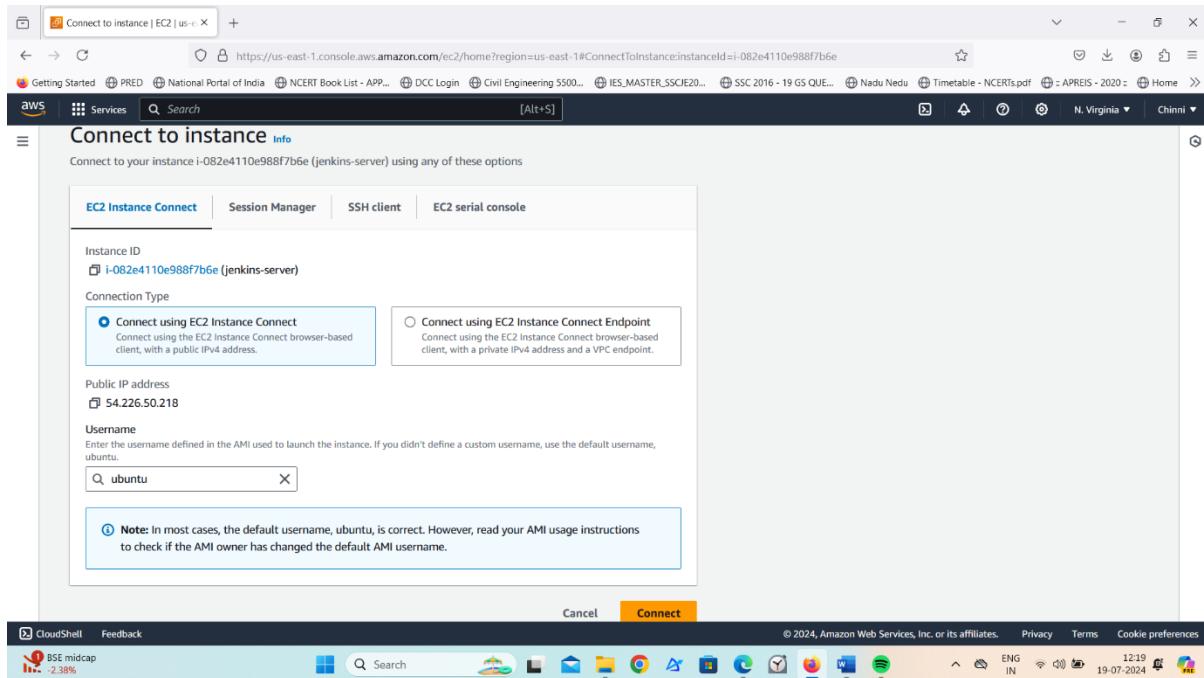
EC2 > Instances > i-082e4110e988f7b6 (jenkins-server) info

Updated less than a minute ago

Instance ID: i-082e4110e988f7b6 (jenkins-server)	Public IPv4 address: 54.226.50.218 [open address]	Private IPv4 addresses: 172.31.19.92
IPv6 address: -	Instance state: Running	Public IPv4 DNS: ec2-54-226-50-218.compute-1.amazonaws.com [open address]
Hostname type: IP name: ip-172-31-19-92.ec2.internal	Private IP DNS name (IPv4 only): ip-172-31-19-92.ec2.internal	Elastic IP addresses: -
Answer private resource DNS name: IPv4 (A)	Instance type: t2.2xlarge	AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. [Learn more]
Auto-assigned IP address: 54.226.50.218 [Public IP]	VPC ID: vpc-0ef979d7194b2891f	Auto Scaling Group name: -
IAM Role: -	Subnet ID: subnet-0a60c45178eb227d4	
IMDSv2 Required	Instance ARN: arn:aws:ec2:us-east-1:891377277875:instance/	

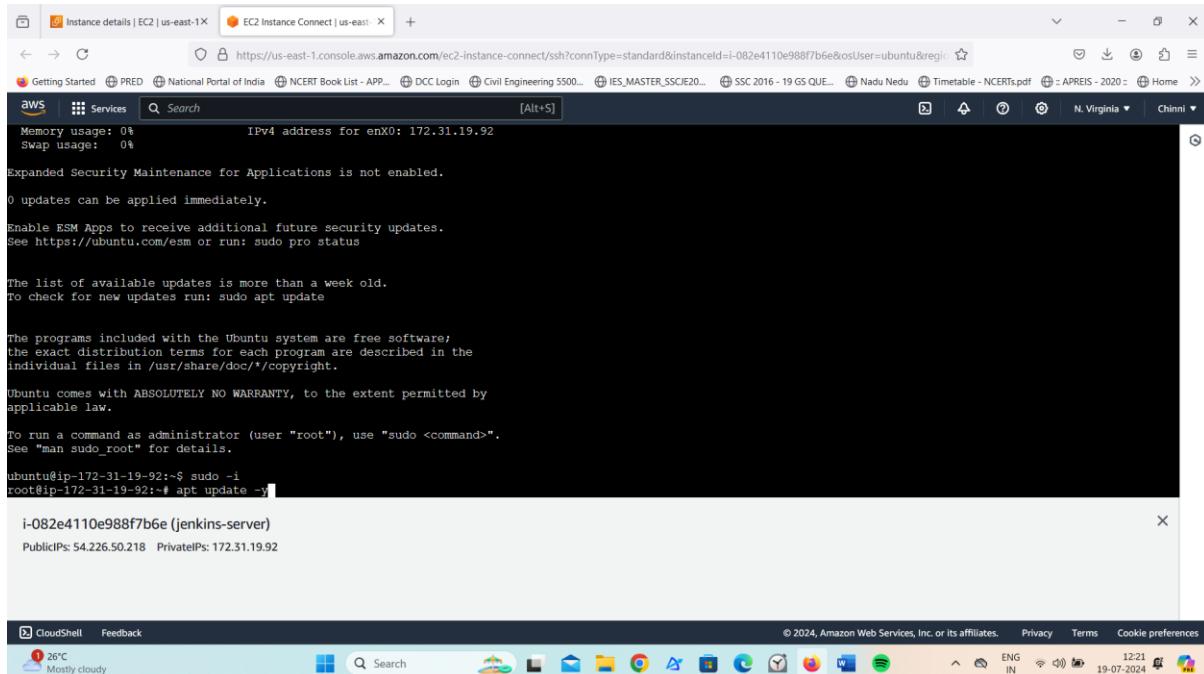
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 12:19 19-07-2024

Connect to the instance.



sudo -i → convert to root user

update the server → apt update -y



Step2: Create IAM role

Search for IAM in the search bar of AWS and click on roles.

Click on Create Role

Select entity type as AWS service

The screenshot shows the 'Create role' wizard in the AWS IAM console. The current step is 'Select trusted entity'. In the 'Trusted entity type' section, the 'AWS service' option is selected. Other options like 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy' are also available. Below this, the 'Use case' section is shown, with 'EC2' selected from a dropdown menu. The browser's address bar shows the URL: https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create.

Use case as EC2 and click on Next.

The screenshot shows the 'Create role' wizard in the AWS IAM console, currently at Step 2: 'Use case'. The 'Service or use case' dropdown is set to 'EC2'. The 'Choose a use case for the specified service' section lists several options under the 'EC2' heading. The 'EC2' option is selected. Other listed options include 'EC2 Role for AWS Systems Manager', 'EC2 Fleet Role', 'EC2 - Spot Fleet Auto Scaling', 'EC2 - Spot Fleet Tagging', 'EC2 - Spot Instances', 'EC2 - Spot Fleet', and 'EC2 - Scheduled Instances'. The browser's address bar shows the URL: https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create.

For permission policy select Administrator Access, click Next.

The screenshot shows the 'Add permissions' step of the 'Create role' wizard. Under 'Permissions policies (1/943)', the 'AdministratorAccess' policy is selected. Other policies listed include 'AdministratorAccess-Amplify', 'AdministratorAccess-AWSElasticBeanstalk', 'AlexaForBusinessDeviceSetup', 'AlexaForBusinessFullAccess', 'AlexaForBusinessGatewayExecution', 'AlexaForBusinessFileSizeDelegatedAccess...', and 'AlexaForBusinessPolicyDelegatedAccess...'. The interface includes a search bar, a filter by type dropdown, and a pagination control.

Provide a Name for Role and click on Create role.

The screenshot shows the 'Role details' section of the 'Create role' wizard. The 'Role name' field contains 'uber-role'. Below it, the 'Description' field contains 'Allows EC2 instances to call AWS services on your behalf.' The 'Step 1: Select trusted entities' section shows the JSON trust policy:

```

1: {
2:   "Version": "2012-10-17",
3:   "Statement": [
4:     {
5:       "Effect": "Allow",
6:       "Action": [
7:         "sts:AssumeRole"
8:       ],
9:       "Principal": [
10:         {
11:           "Service": [
12:             "ec2.amazonaws.com"
13:           ]
14:         }
15:       ]
16:     }
17:   ]
18: }

```

The screenshot shows the 'Step 2: Add permissions' section of the 'Create role' wizard. The 'Permissions policy summary' table lists the attached policy:

Policy name	Type	Attached as
AdministratorAccess	AWS managed - job function	Permissions policy

Role is created.

The screenshot shows the AWS IAM Roles page. The 'uber-role' is selected. The 'Summary' section displays the creation date (July 19, 2024), ARN (arn:aws:iam::891377277875:role/uber-role), and instance profile ARN (arn:aws:iam::891377277875:instance-profile/uber-role). The 'Permissions' tab is active, showing a search bar and a list of managed policies. The status bar at the bottom indicates it's from 19-07-2024, 12:42 AM, and the user is in N. Virginia.

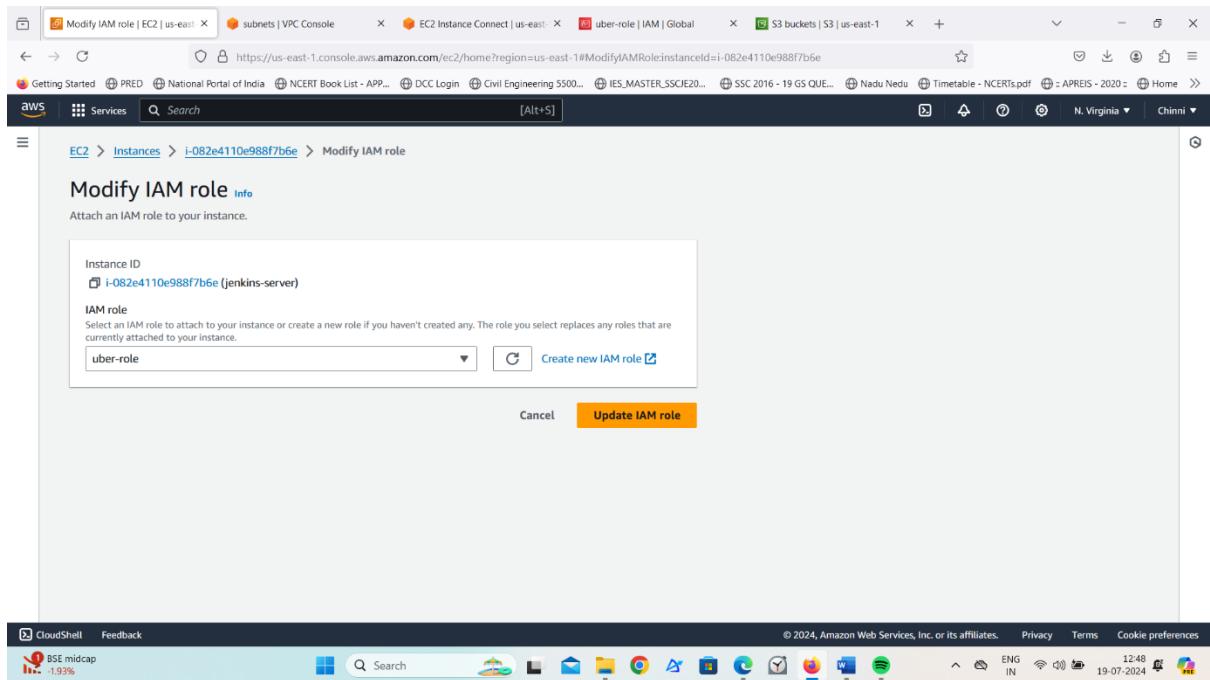
Now Attach this role to Ec2 instance that we created earlier, so we can provision cluster from that instance.

Go to EC2 Dashboard and select the instance.

Click on Actions → Security → Modify IAM role.

The screenshot shows the AWS EC2 Instances page. The 'jenkins-server' instance (i-082e4110e988f7b6e) is selected. The 'Actions' menu is open, showing options like Connect, View details, Manage instance state, Instance settings, Networking, Security, Image and templates, and Monitor and troubleshoot. The status bar at the bottom indicates it's from 19-07-2024, 12:48 AM, and the user is in N. Virginia.

Select the Role that created earlier and click on Update IAM role.



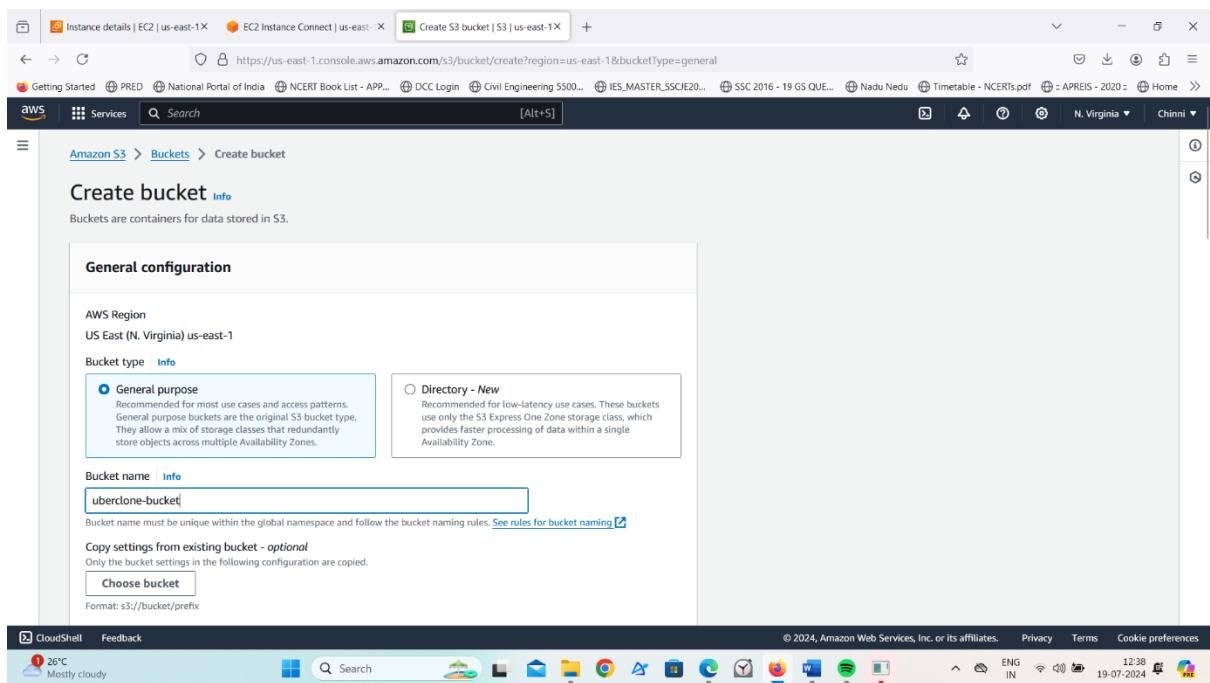
Connect to the instance.

Step3: Create S3 bucket

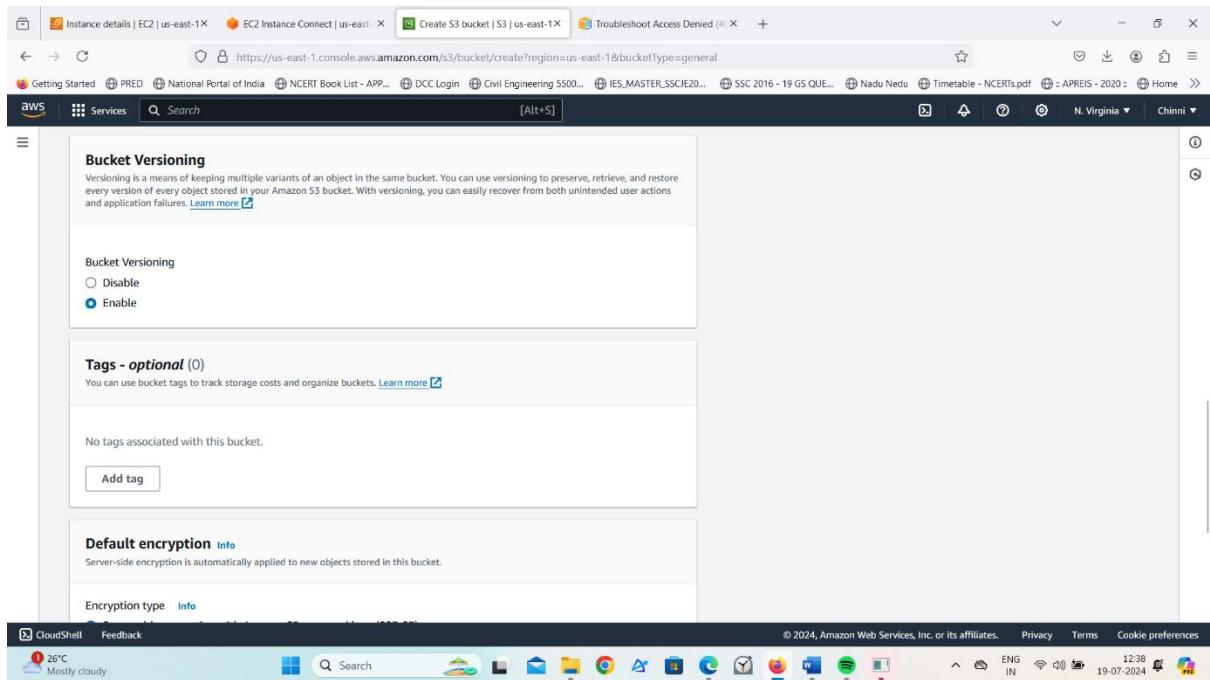
Search for S3 bucket in the search bar of AWS and click on it.

Click on S3 bucket.

Choose bucket type as general purpose and give bucket name.

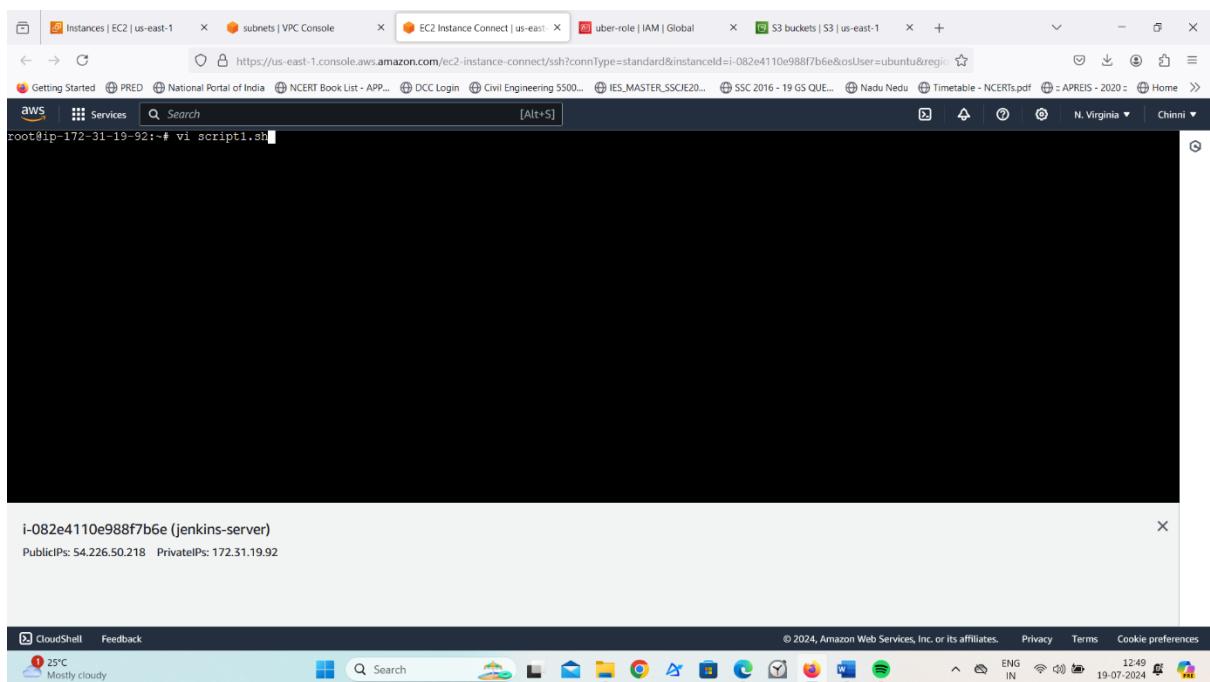


Enable bucket versioning. Click on create bucket.



Step4: Installation of packages

- Create shell script in our server. Using vi script1.sh
- Enter this script into it. This script installs Java, Jenkins, maven, Docker.



```

#!/bin/bash
apt install default-jdk -y
apt install maven -y
#install jenkins
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/" | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
#install docker
apt install docker.io -y

```

-- INSERT --

i-082e4110e988f7b6e (jenkins-server)

PublicIPs: 54.226.50.218 PrivateIPs: 172.31.19.92

For that file change the file permission, using chmod. And execute the file.

```

root@ip-172-31-19-92:~# vi script1.sh
root@ip-172-31-19-92:~# chmod u+x script1.sh
root@ip-172-31-19-92:~# ./script1.sh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsu-ucm-conf at-spi2-common at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service default-jdk-headless
  default-jre default-jre-headless fontconfig fontconfig-config fonts-dejavu-core fonts-dejavu-extra fonts-dejavu-mono gsettings-desktop-schemas
  gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme java-common libasound2-data libasound2d4 libatk-biffge2.0-0t64 libatk-wrapper-java
  libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64 libavahi-client3 libavahi-common-data libavahi-common libcairo-gobject libcairo2 libcolor2
  libcupsc2t64 libdconf1 libdeflate0 libdrm-intel libdrm-nouveau2 libdrm-radeon libepoxy libfrontconfig libgd-pixbuf-2.0-0
  libgdk-pixbuf2.0-0t64 libgtk-3-bin libgtk-3-common libhashtuzz0b libice6 libjbig2 libjpeg-turbo8 libjpeg8 liblcms2-2 liblerc4 liblwm1764 libpango-1.0-0
  libpangocairo-1.0-0 libpangoft2-1.0 libpiciaccesso libpixman-1.0 libpthread-stubs0-dev librsvg2-2 librsvp2-common libsharpuyuv0 libsm6
  libthal-data libthal0 libxf6 libvulkan libwayland-cursor0 libwebp7 libx11-dev libx11-xcb libxau-dev libxaw7 libxcb-dri2-0
  libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-render0 libxcb-shape0 libxcb-sync libxcb-xfixes0 libxcb-dri2 libxcb-composite0
  libxcursor libximage0 libximcp-dev libxixxes3 libxf86 libxineramapi libxkbfile libxmu libxpme libxiandri libxsmmencel libxt-dev libxt64
  libxtst6 libxv1 libxf86dg1 libxxf86vm mesa-vulkan-drivers openjdk-21-jdk openjdk-21-jre-headless openjdk-21-jre openjdk-21-jre-headless session-migration
  ubuntuutils x11-common x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  alsu-ucm libasound2-plugins colord cups-common gvfs libice-doc liblcm2s-0-utils pcscd librsvg2-bin libssm-doc libx11-doc libxcb-doc libxt-doc openjdk-21-demo
  openjdk-21-source visualvm libnss-mins fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zhenhei fonts-indic mesa-utils
Recommended packages:
  iut
The following NEW packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsu-ucm-conf at-spi2-common at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service default-jdk

i-082e4110e988f7b6e (jenkins-server)
PublicIPs: 54.226.50.218 PrivateIPs: 172.31.19.92

```

Create another script file using vi script2.sh

This script installs Kubectl, Terraform, AWS Cli, SonarQube, Trivy.

For that file change the file permission, using chmod. And execute the file.

```
'/bin/bash
#install trivy
sudo apt-get install wget apt-transport-https gnupg lsb-release
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null
echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb $ (lsb_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list
sudo apt-get update
sudo apt-get install trivy
apt install unzip -y
#install awscli
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
#terraform
wget -qO- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
sudo apt update && sudo apt install terraform
#kubectl
sudo apt update
sudo apt install curl -y
curl -L https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl
-- INSERT --
```

i-082e4110e988f7b6e (jenkins-server)
PublicIPs: 54.226.50.218 PrivateIPs: 172.31.19.92



```
root@ip-172-31-19-92:~# vi script2.sh
root@ip-172-31-19-92:~# chmod u+x script2.sh
root@ip-172-31-19-92:~# ./script2.sh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.4-1ubuntu4.1).
wget set to manually installed.
gnupg is already the newest version (2.4.4-2ubuntu17).
gnupg set to manually installed.
lsb-release is already the newest version (12.0-2).
lsb-release set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 22 not upgraded.
Need to get 3974 B of archives.
After this operation, 35.8 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Fetched 3974 B in 0s (238 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 84671 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Setting up apt-transport-https (2.7.14build2) ...
scanning processes...
scanning linux images...
```

i-082e4110e988f7b6e (jenkins-server)
PublicIPs: 54.226.50.218 PrivateIPs: 172.31.19.92



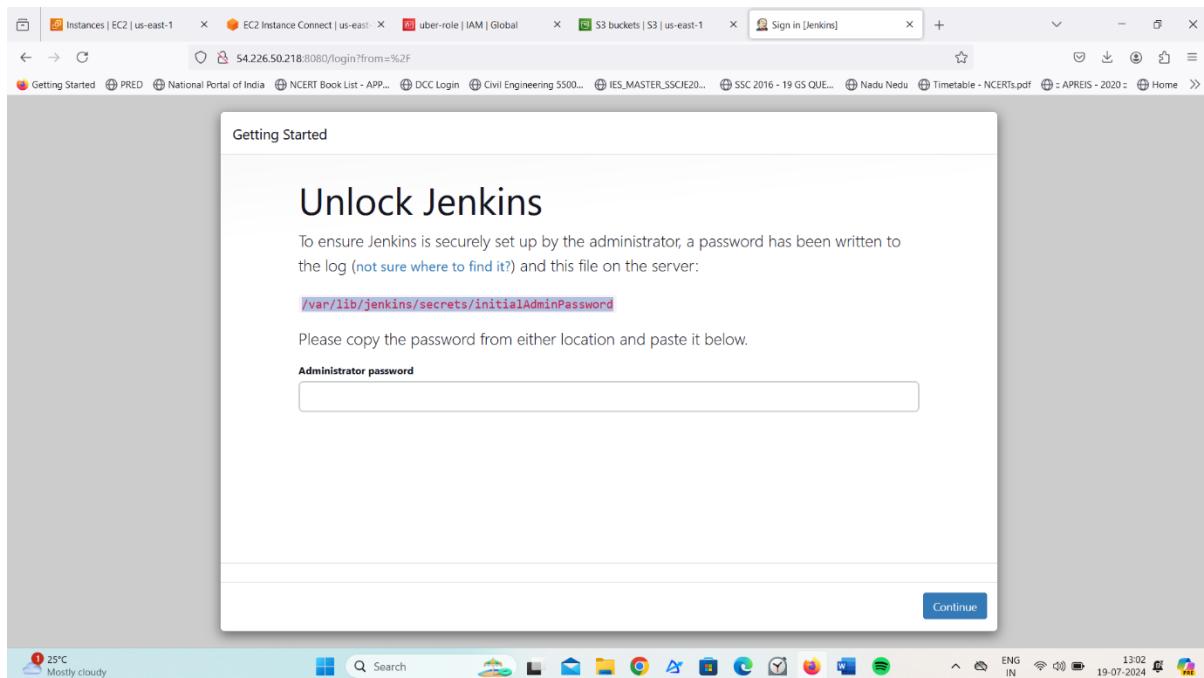
Step5: Connect to Jenkins and SonarQube servers

Now copy the public IP address of ec2 and paste it into the browser

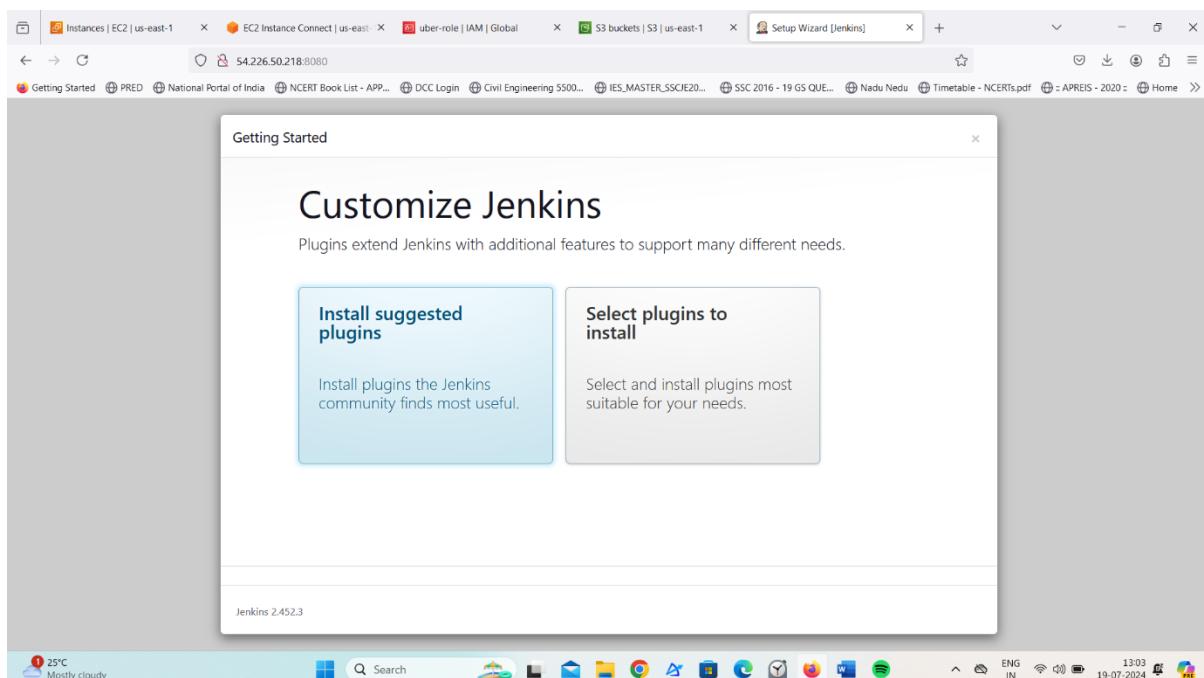
<Ec2-ip:8080> #you will see Jenkins login page

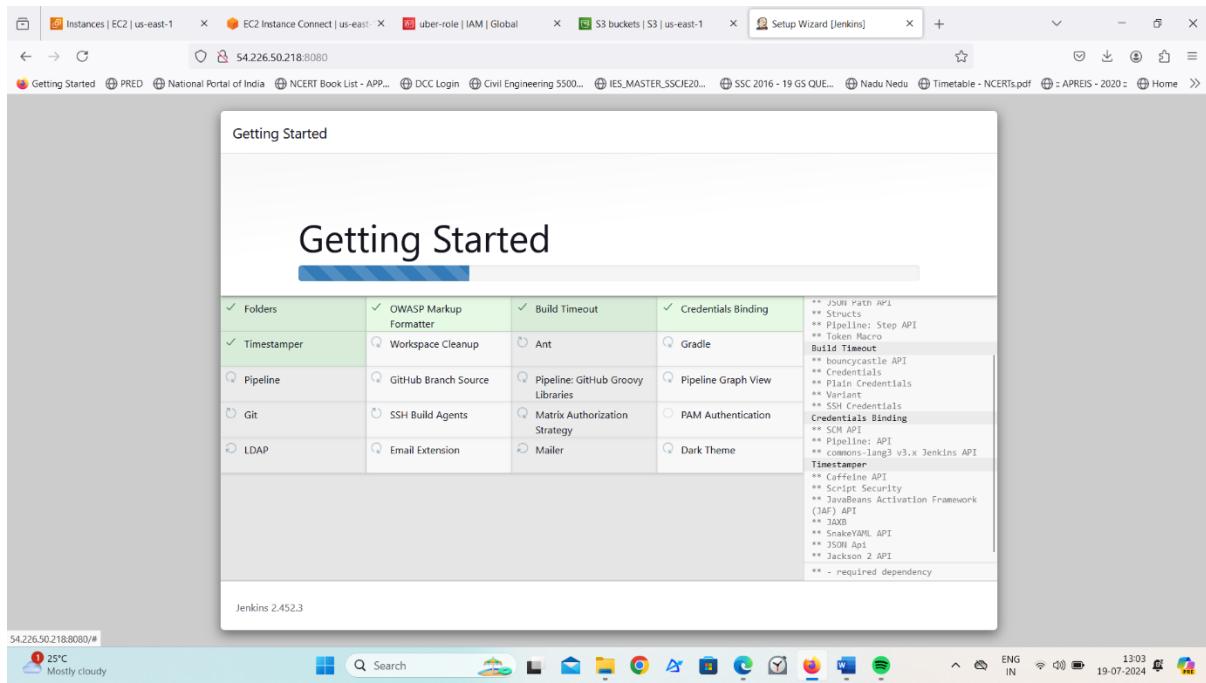
provide the below command for the Administrator password

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```



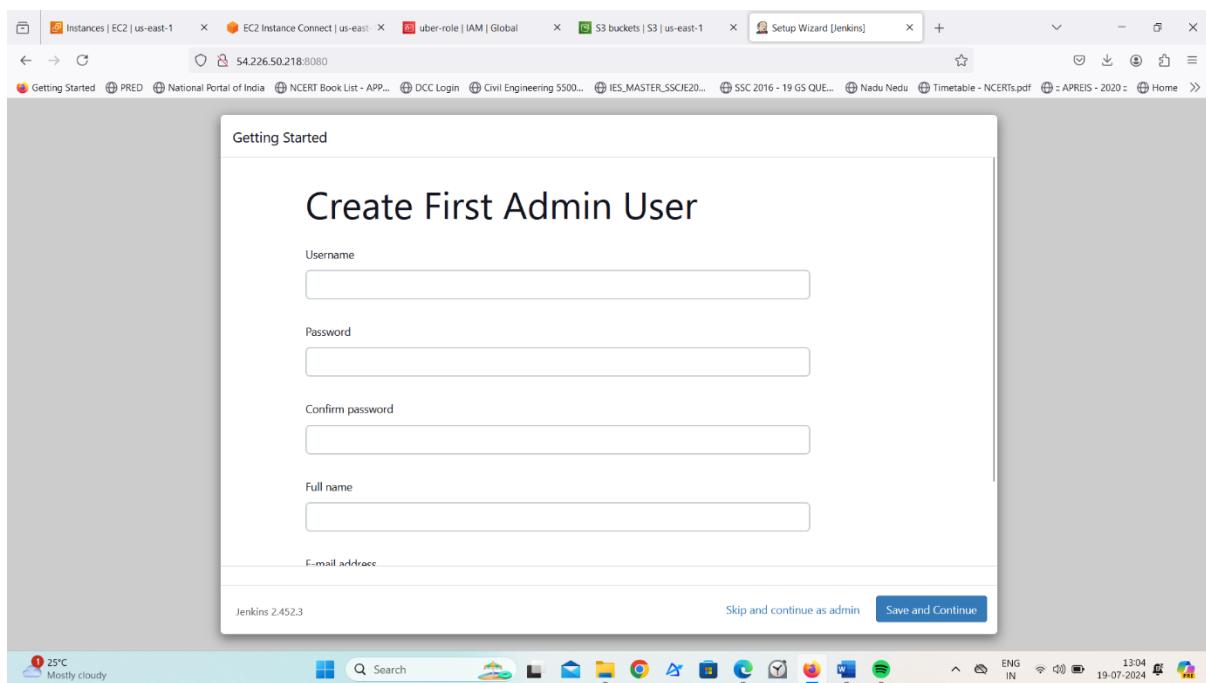
Now, install the suggested plugins.



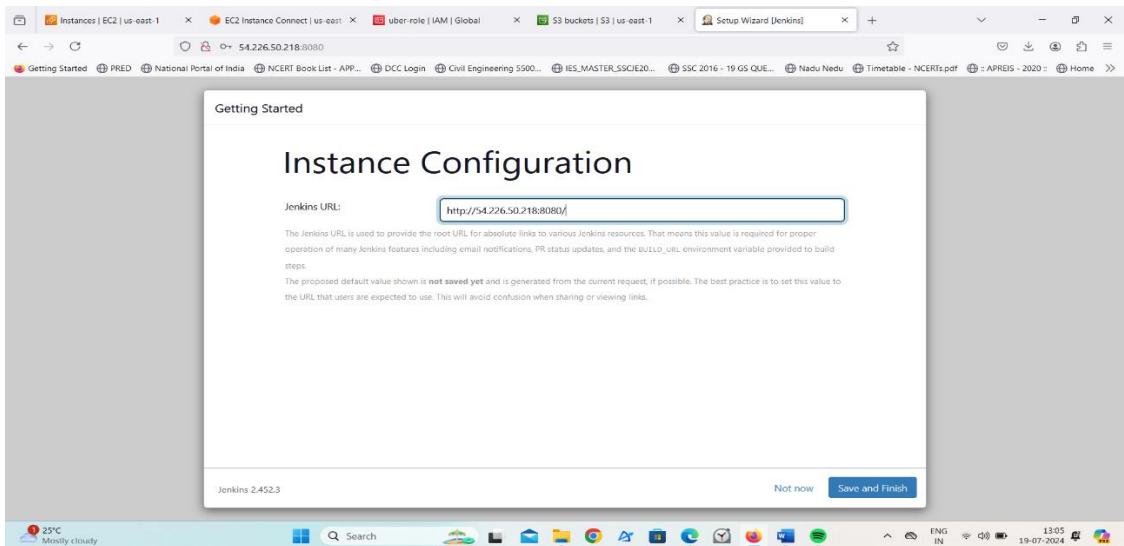


Jenkins will now get installed and install all the libraries.

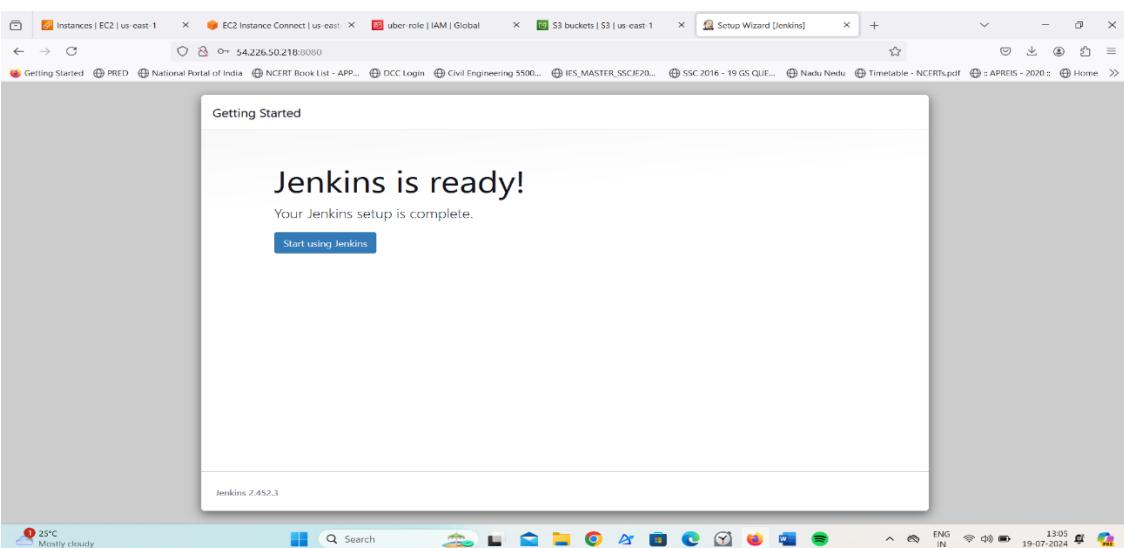
Create an admin user



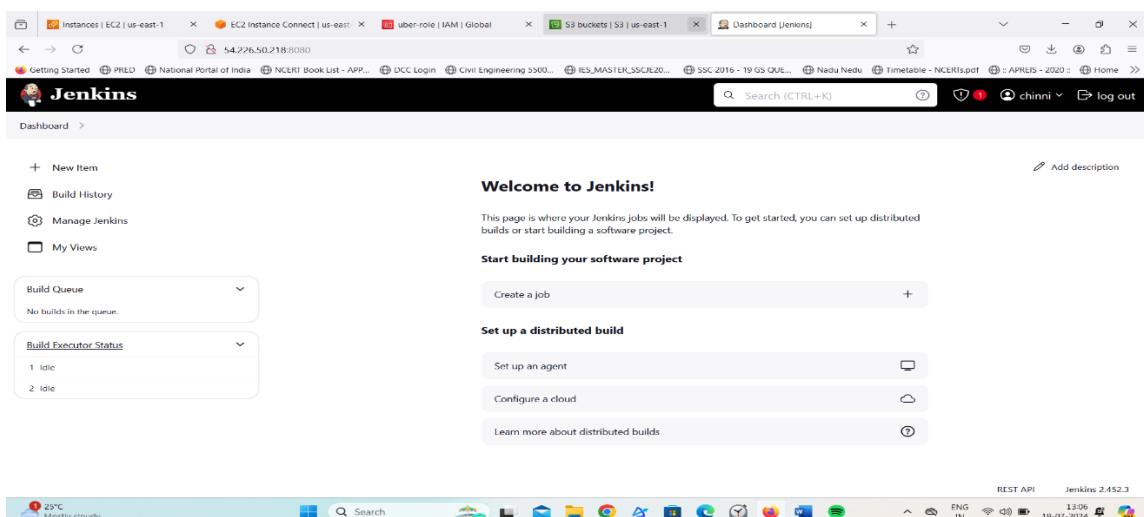
Click on save and continue.



Click on save and Finish



Click on start using Jenkins. We see Jenkins dashboard.



Using docker we will install SonarQube server.

```
sudo chmod 777 /var/run/docker.sock
```

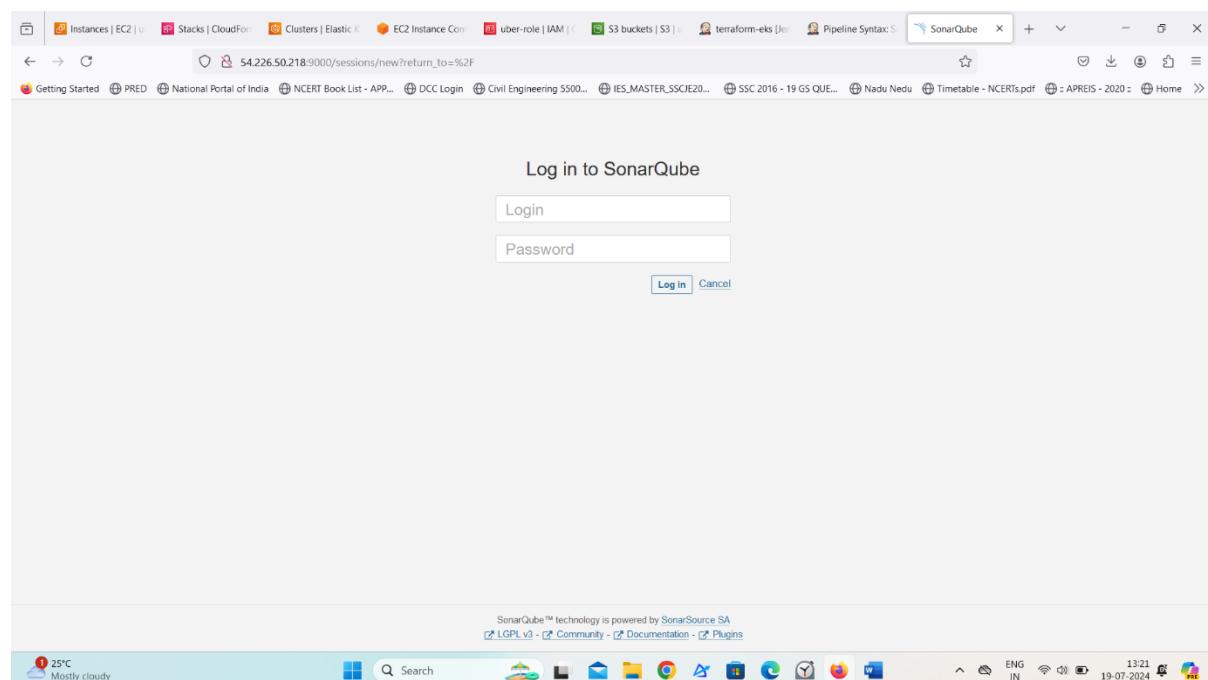
```
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```

```
$ Total % Received % Xferd Average Speed Time Time Current  
          Total Upload Total Spent Left Speed  
100 138 100 138 0 0 2074 0 ---:--- ---:--- ---:--- 2090  
100 49.0M 100 49.0M 0 0 92.6M 0 ---:--- ---:--- ---:--- 92.6M  
Client Version: v1.30.3  
Kubernetes Version: v1.23.11-eks-165947-6ce0bf390ce3  
root@ip-172-31-19-92:~# cat  
  
root@ip-172-31-19-92:~# cat /var/lib/jenkins/secrets/initialAdminPassword  
2b39f6a939794e2497bc5d8c7dab2d53  
root@ip-172-31-19-92:~# sudo chmod 777 /var/run/docker.sock  
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community  
Unable to find image 'sonarqube:lts-community' locally  
lts-community: Pulling from library/sonarqube  
9b857f539cb1: Pull complete  
708ff3b02f8b: Pull complete  
elsea69141092: Pull complete  
5d591beb7c55: Pull complete  
chb850744c92: Pull complete  
4bbbe66c34eb: Pull complete  
2d55993c554: Pull complete  
4f4f700ef54: Pull complete  
Digest: sha256:f51d604ae94717faf2bf5c63cb90429d445f787ff2ac33ada38c0a36d8f8afe  
Status: Downloaded newer image for sonarqube:lts-community  
2b773e63e93b8ed2af94d02e0baac71c97f35f5d5d679cc9abcb7839a2e47e0  
root@ip-172-31-19-92:~#  
  
i-082e4110e988f7b6e (jenkins-server)  
Public IPs: 54.226.50.218 Private IPs: 172.31.19.92
```

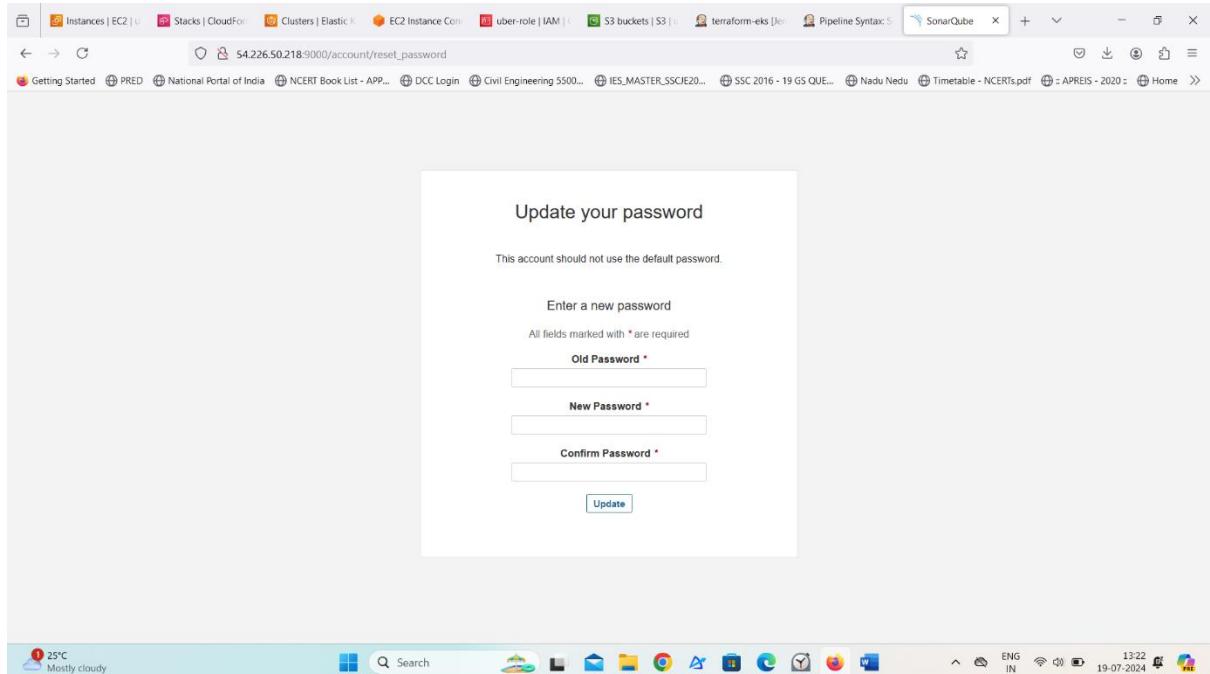
The screenshot shows a CloudShell terminal window within the AWS Management Console. The terminal output displays the execution of Docker commands to run the SonarQube container. It includes the command to change file permissions, the Docker run command, and the resulting error message about the image not being found locally. The terminal also shows the IP address of the Jenkins server (i-082e4110e988f7b6e) and its public and private IP addresses.

Now Copy the public IP again and paste it into a new tab in the browser with 9000

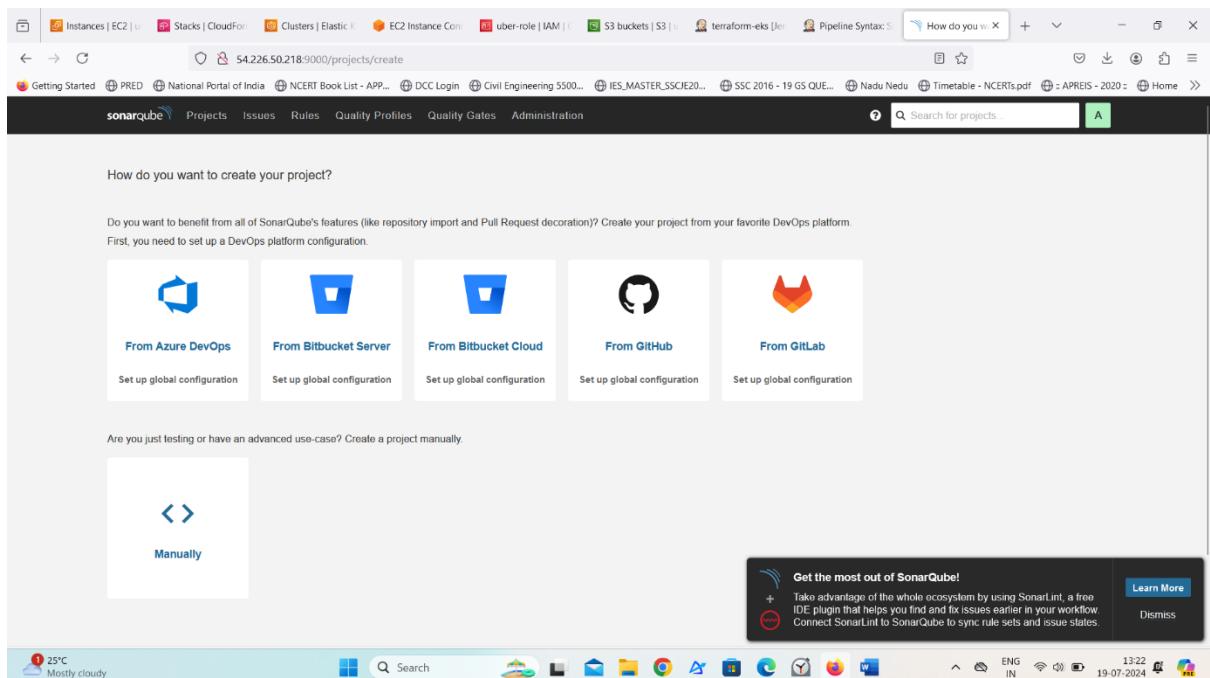
```
<ec2-ip:9000> #runs sonar container
```



Enter username and password, click on login and change password
username admin
password admin



Update New password, this is Sonar Dashboard.



We see sonar server is accessed.

Step6: Terraform plugin install and EKS provision

Now go to Jenkins and add a terraform plugin to provision the AWS EKS using the Pipeline Job.

Go to Jenkins dashboard → Manage Jenkins → Plugins

Available Plugins, Search for Terraform and install it.

The screenshot shows the Jenkins Manage Jenkins interface. In the left sidebar, 'Manage Jenkins' is selected. The main area is titled 'System Configuration' and contains sections for 'System', 'Tools', 'Plugins', 'Nodes', 'Clouds', and 'Appearance'. A message at the top right says: 'Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.' Below the configuration sections, there's a 'Security' section with links to 'Security', 'Credentials', and 'Credential Providers'.

The screenshot shows the Jenkins Manage Jenkins interface with the 'Plugins' section selected in the left sidebar. The main area displays a list of available plugins under the 'Updates' tab. A message at the bottom states: 'Disabled rows are already upgraded, awaiting restart. Shaded but selectable rows are in progress or failed.' The Jenkins header includes a weather widget showing '25°C Mostly cloudy'.

I also installed pipeline stage view to see our stages.

let's find the path to our Terraform (we will use it in the tools section of Terraform)

which terraform

Now come back to Manage Jenkins → Tools

Add the terraform in Tools

Ant installations

Add Ant

Maven installations

Add Maven

Terraform installations

Add Terraform

Save Apply

Jenkins 2.452.3

Give name for it and give install directory (where we installed terraform)

Ant installations

Add Ant

Maven installations

Add Maven

Terraform installations

Add Terraform

Name
terraform

Install directory
/usr/bin/

Install automatically ?

Add Terraform

Save Apply

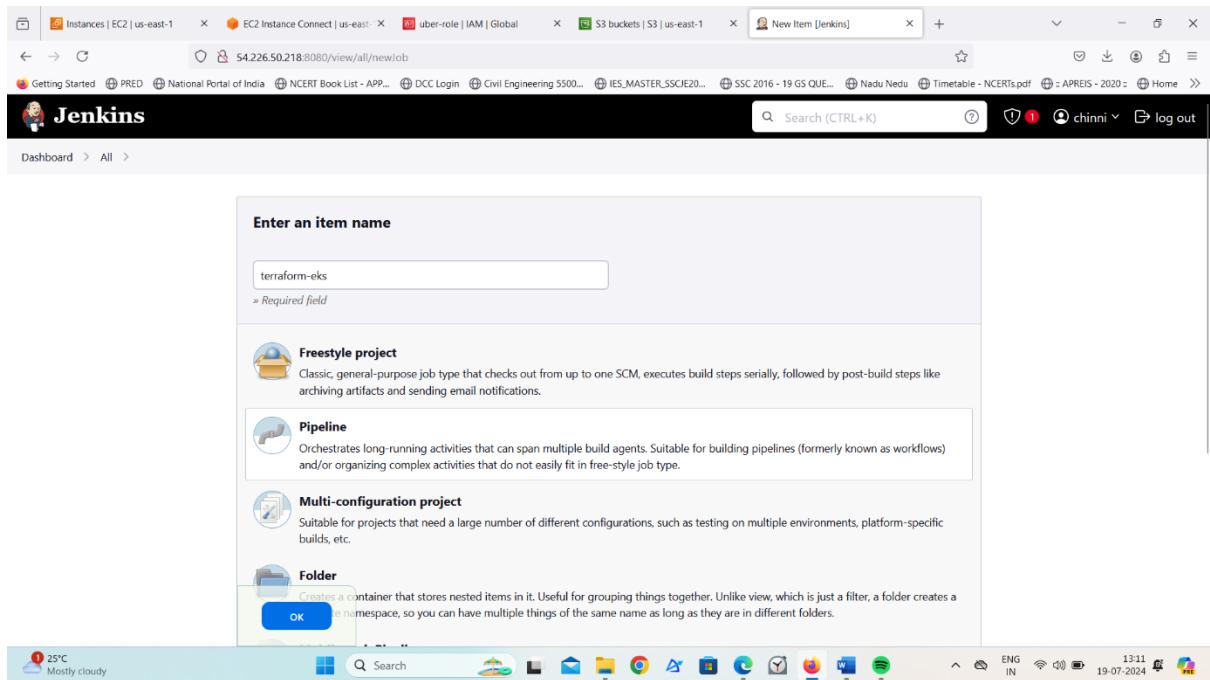
Jenkins 2.452.3

Click on apply and then save it

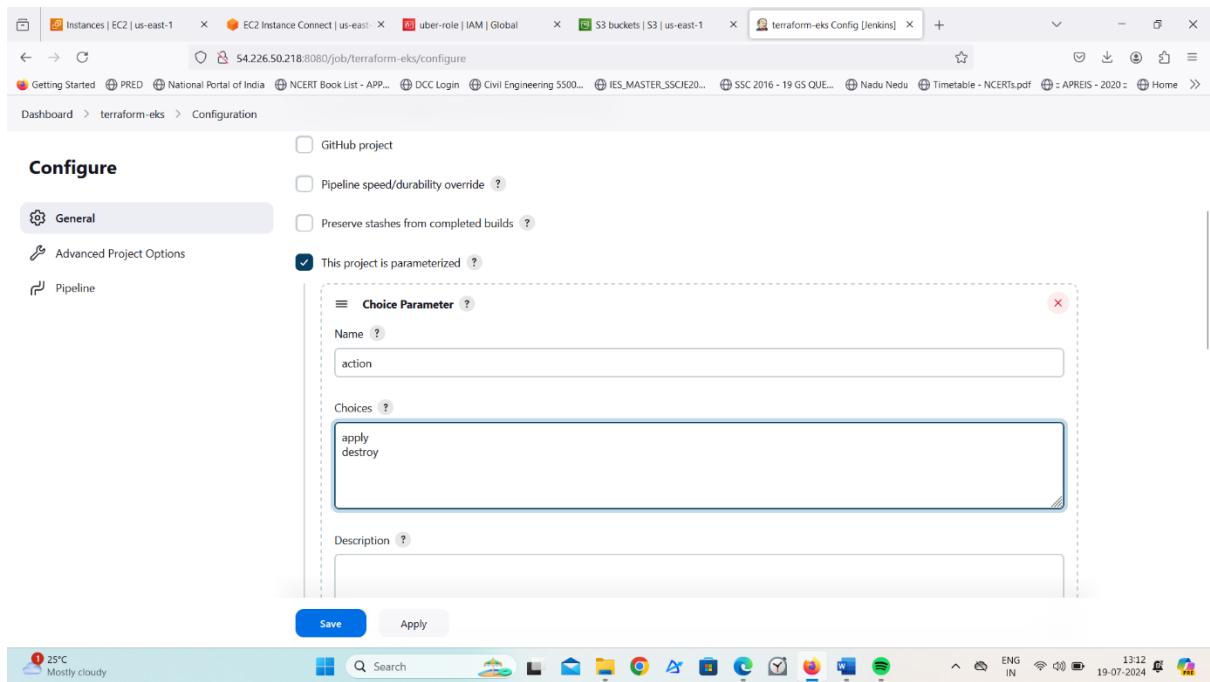
Now go to Jenkins dashboard.

Create a new job for the EKS provision

Click on new item.



I want to do this with build parameters to apply and destroy while building only.
you have to add this inside job like the below image



Before adding pipeline script, we have to do some changes in terraform configuration file.

Go to git hub and change bucket name in backend.tf file.

```

1 terraform {
2   backend "s3" {
3     bucket = "uberclone-bucket" # Replace with your actual S3 bucket name
4     key    = "EKS/terraform.tfstate"
5     region = "us-east-1"
6   }
7 }

```

Also change role_arn, subnet ids, security group id in main.tf file.

```

1 resource "aws_eks_cluster" "my_cluster" {
2   name      = "my-cluster"
3   role_arn  = "arn:aws:iam::89137727785:role/uber-role" # Replace with your IAM role ARN
4
5   vpc_config {
6     subnet_ids       = ["subnet-0a60c45178eb227d4", "subnet-02d78eb93acef862f"] # Replace with your subnet IDs
7     security_group_ids = ["sg-02b97bab5a63fec8a"]                                # Replace with your security group IDs
8   }
9
10  tags = {
11    Environment = "Production"
12  }
13
14
15 resource "aws_eks_node_group" "my_node_group" {
16   cluster_name  = aws_eks_cluster.my_cluster.name
17   node_group_name = "my-node-group"
18   node_role_arn = "arn:aws:iam::89137727785:role/uber-role" # Replace with your IAM role ARN
19   subnet_ids       = ["subnet-0a60c45178eb227d4", "subnet-02d78eb93acef862f"] # Replace with your subnet IDs
20
21   scaling_config {
22     desired_size = 1
23     max_size     = 2
24     min_size     = 1
25   }
26 }

```

In provider.tf change your region.

```

pipeline {
  agent any
  stages {
    stage ('Checkout from Git') {
      steps {
        checkout scmGit(branches: [[name: '/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/vommidapuchinni/uber-clone.git']])
      }
    }
  }
}

```

```
}

stage ('Terraform version'){

steps{
    sh 'terraform --version'

}

}

stage('Terraform init'){

steps{
    dir('EKS_TERRAFORM') {
        sh 'terraform init'

    }

}

stage('Terraform validate'){

steps{
    dir('EKS_TERRAFORM') {
        sh 'terraform validate'

    }

}

stage('Terraform plan'){

steps{
    dir('EKS_TERRAFORM') {
        sh 'terraform plan'

    }

}

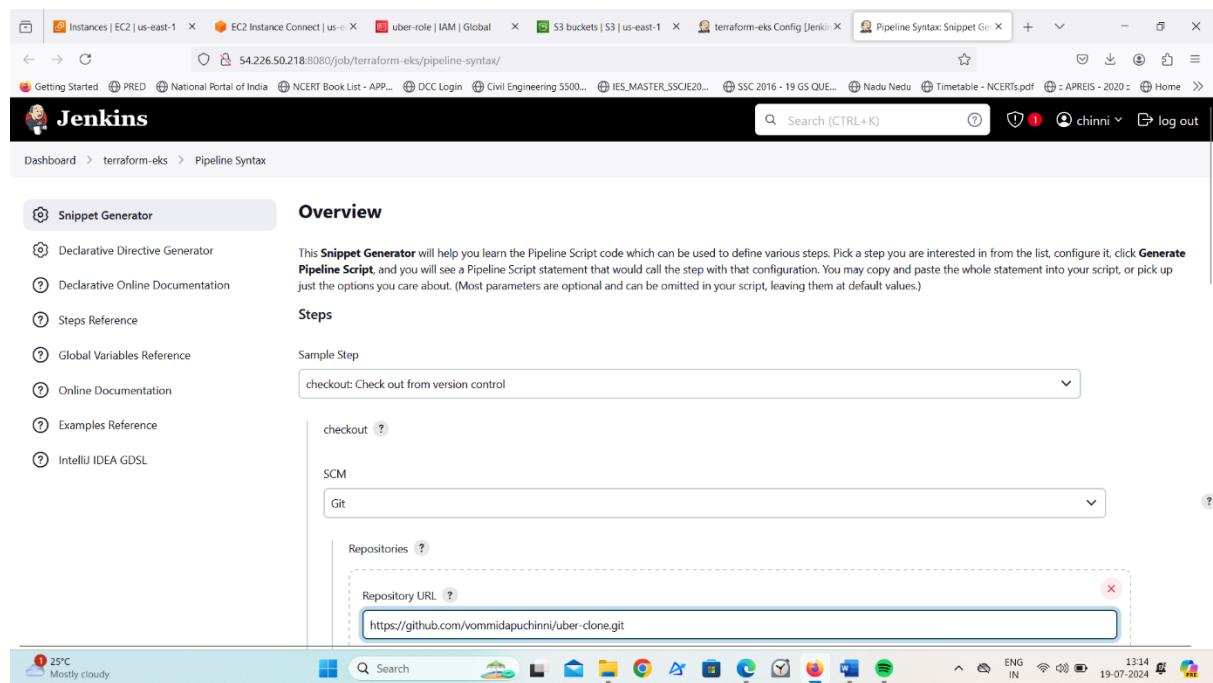
}
```

```
stage('Terraform apply/destroy'){

    steps{
        dir('EKS_TERRAFORM') {
            sh 'terraform ${action} --auto-approve'
        }
    }

}

}
```



From this pipeline syntax we get our checkout syntax.

Click on apply and then save.

Click on build parameters.

It will ask for actions to apply/destroy.

Choose apply and click on build.

Jenkins

Pipeline terraform-eks

Status Changes Build with Parameters Configure Delete Pipeline Full Stage View Stages Rename Pipeline Syntax

This build requires parameters:

action

apply

Build Cancel

Build History trend ▾

No builds

25°C Mostly cloudy

It will take few minutes to create it.

First, I got error to create it because I didn't give permission to create node group.

Now go to its IAM role, under trust relationships, click on edit trust policy.

Give eks permission along with ec2.

Services Search [Alt+S]

IAM Roles uber-role Edit trust policy

CloudShell Feedback

CloudShell Feedback

26°C Mostly cloudy

Click on save policy.

The screenshot shows the AWS IAM console with the 'uber-role' selected. The 'Trust relationships' tab is active, displaying the JSON policy document:

```

1  [
2    {
3      "Version": "2012-10-17",
4      "Statement": [
5        {
6          "Effect": "Allow",
7          "Principal": "*",
8          "Service": [
9            "eks.amazonaws.com",
10           "ec2.amazonaws.com"
11         ],
12         "Action": "sts:AssumeRole"
13     }
14   ]
15 ]

```

We see like this under IAM role.

After editing it come to Jenkins server again click on build with parameters choose apply and click on build.

The screenshot shows the Jenkins Pipeline Syntax stage view for the 'terraform-eks' pipeline. The table displays the following data:

	Checkout from Git	Terraform version	Terraform init	Terraform validate	Terraform plan	Terraform apply/destroy
#3 Jul 19, 2024, 8:08 AM	277ms	361ms	3s	3s	3s	2min 0s
#2 Jul 19, 2024, 8:07 AM	267ms	358ms	4s	3s	3s	4s failed
#1 Jul 19, 2024, 7:45 AM	3s	399ms	7s	3s	3s	8min 58s failed

Jenkins Dashboard

S	W	Name	Last Success	Last Failure	Last Duration
		terraform eks	35 min #3	37 min #2	2 min 13 sec

Build Queue: No builds in the queue.

Build Executor Status: 1 Idle, 2 Idle

Windows Taskbar: Jenkins icon, URL 54.226.50.218:8080/view/all/newJob

Go to AWS console we see our cluster is created and one node group also.

Amazon Elastic Kubernetes Service

Clusters (1) info

Cluster name	Status	Kubernetes version	Support period	Created	Provider
my-cluster	Active	1.30	Standard support until July 28, 2025	28 minutes ago	EKS

Windows Taskbar: Jenkins icon, URL https://us-east-1.console.aws.amazon.com/eks/home?region=us-east-1#clusters

EC2 Dashboard

Instances (1/2) info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
jenkins-server	i-082e4110e988f7b6e	Running	t3.medium	2/2 checks passed	View alarms +	us-east-1a

i-082e4110e988f7b6e (jenkins-server)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Public IPv4 address: 54.226.50.218 | Open address

Private IPv4 addresses: 172.31.19.92

Step7: Plugins installation & setup (Java, Sonar, Nodejs, OWASP, Docker)

Go to Jenkins dashboard

Manage Jenkins → Plugins → Available Plugins

Search for the Below Plugins

Eclipse Temurin installer

SonarQube Scanner

NodeJS

OWASP Dependency-Check

Docker

Docker Commons

Docker Pipeline

Docker API

Docker-build-step

The screenshot shows the Jenkins 'Available plugins' page. The top navigation bar includes links for Instances | EC2, Stacks | CloudFormation, Clusters | Elastic, EC2 Instance Connect, IAM, S3 buckets | S3, Available plugins, Pipeline Syntax: S, How do you want to use Jenkins, Home, and Log Out. The main content area has a search bar labeled 'Search (CTRL+K)' and a 'Install' button. On the left, there's a sidebar with 'Updates', 'Available plugins' (which is selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main table lists the following plugins:

Install	Name	Released	
<input checked="" type="checkbox"/>	Eclipse Temurin installer 1.5	Provides an installer for the JDK tool that downloads the JDK from https://adoptium.net	1 yr 9 mo ago
<input checked="" type="checkbox"/>	SonarQube Scanner 2.17.2	External Site/Tool Integrations Build Reports	5 mo 0 days ago
<input checked="" type="checkbox"/>	NodeJS 1.6.1	npm	11 mo ago
<input checked="" type="checkbox"/>	OWASP Dependency-Check 5.5.1	Security DevOps Build Tools Build Reports	13 days ago
<input checked="" type="checkbox"/>	Docker 1.6.2	Cloud Providers Cluster Management docker	1 mo 15 days ago

The status bar at the bottom shows the weather as '27°C Mostly cloudy', system icons, and the date/time as '13:47 19-07-2024'.

The screenshot shows the Jenkins 'Plugins' page. On the left, there's a sidebar with links like 'Updates', 'Available plugins' (which is highlighted), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area has a search bar at the top. A list of available plugins is shown, each with a checkbox, name, version, and a brief description. One plugin, 'Docker Pipeline', is checked and highlighted with a yellow background. It has a note below it: 'This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.' At the bottom right of the list, there's a large blue 'Install' button.

Step8: Configure in Global Tool Configuration

Goto Manage Jenkins → Tools

Search for JDK installation.

Click on add JDK.

The screenshot shows the Jenkins 'Tools' configuration page. At the top, there are dropdown menus for 'Default settings provider' (set to 'Use default maven settings') and 'Default global settings provider' (set to 'Use default maven global settings'). Below these, there's a section for 'JDK installations' with a 'Add JDK' button. Under 'Git installations', there's a form for adding a 'Git' installation with fields for 'Name' (set to 'Default') and 'Path to Git executable'. At the bottom, there are 'Save' and 'Apply' buttons.

Give name and click on install automatically. Click on add installer.

Choose install from adoptium.net, choose version.

The screenshot shows the Jenkins 'JDK installations' configuration page. At the top, there is a navigation bar with various links like Instances | EC2, Stacks | CloudFormation, Clusters | Elastic, EC2 Instance Con..., uber-role | IAM, S3 buckets | S3, Tools [Jenkins] (which is selected), Pipeline Syntax, How do you want..., and Home. Below the navigation bar, the URL is 54.226.50.218:8080/manage/configureTools/. The main content area is titled 'JDK installations'. It has a form with a 'Name' field containing 'jdk17', a checked 'Install automatically' checkbox, and a sub-section titled 'Install from adoptium.net' with a 'Version' dropdown set to 'jdk-17.0.8.1+1'. There is also a 'Save' and 'Apply' button at the bottom.

Come down search NodeJS.

Give name and click on install automatically.

Click on add installer, choose install from nodejs.org.

Choose version.

The screenshot shows the Jenkins 'NodeJS' configuration page. At the top, there is a navigation bar with various links like Instances | EC2, Stacks | CloudFormation, Clusters | Elastic, EC2 Instance Con..., uber-role | IAM, S3 buckets | S3, Tools [Jenkins] (selected), Pipeline Syntax, How do you want..., and Home. Below the navigation bar, the URL is 54.226.50.218:8080/manage/configureTools/. The main content area is titled 'NodeJS'. It has a form with a 'Name' field containing 'node16', a checked 'Install automatically' checkbox, and a sub-section titled 'Install from nodejs.org' with a 'Version' dropdown set to 'NodeJS 16.2.0'. There are checkboxes for 'Force 32bit architecture' and 'Global npm packages to install' (with a note about using -- see npm install -g). There is also a 'Save' and 'Apply' button at the bottom.

Search for SonarQube scanner installation, give name and click on install automatically. Click on add installer choose install from maven central.

SonarQube Scanner installations

Add SonarQube Scanner

SonarQube Scanner

Name: sonar-scanner

Install automatically ?

Install from Maven Central

Version: SonarQube Scanner 6.1.0.4477

Add Installer ▾

Add SonarQube Scanner

Save Apply

- Come down search dependency check installation. Click on add dependency check.
- Give name and click on install automatically, click on add installer.
- Choose install from github.com.

Dependency-Check installations

Add Dependency-Check

Dependency-Check

Name: DP-Check

Install automatically ?

Install from github.com

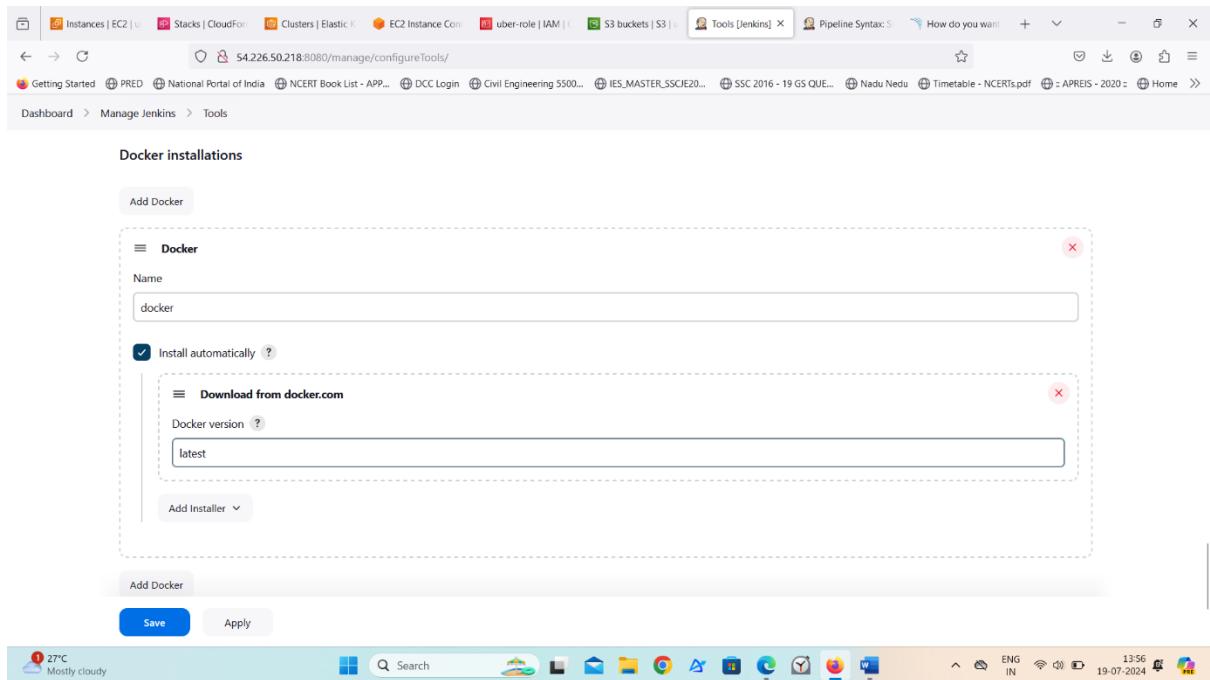
Version: dependency-check 10.0.3

Add Installer ▾

Add Dependency-Check

Save Apply

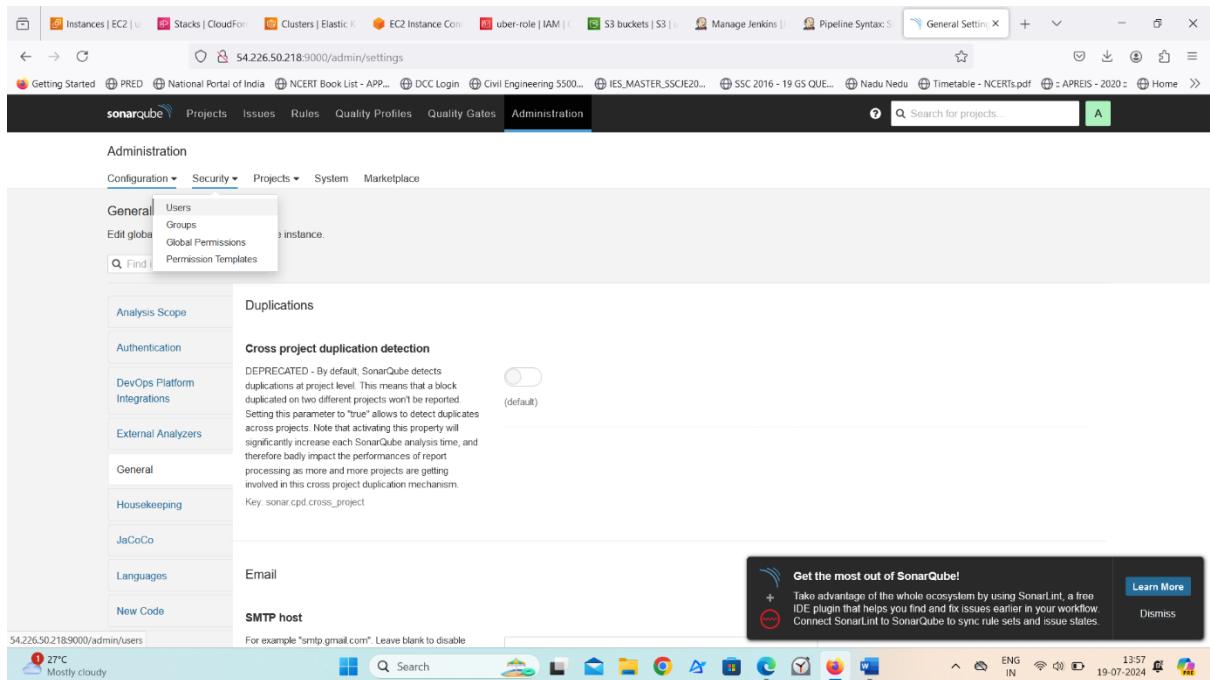
- Come down search for docker installation.
- Click on add docker. Give name. click on install automatically → add installer → choose download from docker.com → choose latest.



Click on apply and then save it.

Step9: Configure Sonar Server in Manage Jenkins

Goto your Sonarqube Server. Click on Administration → Security → Users → Click on Tokens and Update Token → Give it a name → and click on Generate Token.



Create a token with a name and generate

Administration

Configuration ▾ Security ▾ Projects ▾ Issues ▾ Rules ▾ Quality Profiles ▾ Quality Gates ▾ Administration

Users

Create and administer individual users.

Search by login or name:

	SCM Accounts	Last connection	Groups	Tokens
A Administrator admin		< 1 hour ago	sonar-administrators sonar-users	0

1 of 1 shown

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support...
SonarQube™ technology is powered by SonarSource SA
Community Edition - v9.9.6 (build 92038) - [GPL v3](#) - [Community](#) - [Documentation](#)

Get the most out of SonarQube!
Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.
[Learn More](#) [Dismiss](#)

27°C Mostly cloudy 13:57 ENG IN 19-07-2024

Administration

Configuration ▾ Security ▾ Projects ▾ Issues ▾ Rules ▾ Quality Profiles ▾ Quality Gates ▾ Administration

Tokens of Administrator

Generate Tokens

Name	Expires in
<input type="text" value="Enter Token Name"/>	30 days

New token "chinni" has been created. Make sure you copy it now, you won't be able to see it again!

[Copy](#) squ_bbab14daa705d645f588d595a6e6e727f65a6284bb

Name	Type	Project	Last use	Created	Expiration
chinni	User	Never		July 19, 2024	August 18, 2024

[Revoke](#)

Done

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support...
SonarQube™ technology is powered by SonarSource SA
Community Edition - v9.9.6 (build 92038) - [GPL v3](#) - [Community](#) - [Documentation](#)

Get the most out of SonarQube!
Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.
[Learn More](#) [Dismiss](#)

ENG - WI Live 14:06 ENG IN 19-07-2024

copy Token

Goto Jenkins Dashboard → Manage Jenkins → Credentials.

The screenshot shows the Jenkins Manage Jenkins interface. In the top navigation bar, there are several links including 'Instances | EC2', 'Stacks | CloudFormation', 'Clusters | Elastic...', 'EC2 Instance Con...', 'uber-role | IAM', 'S3 buckets | S3', 'Manage Jenkins', 'Pipeline Syntax', 'Users - Administra...', and 'Home'. Below the navigation, a search bar contains the URL '54.226.50.218:8080/manage/'. A breadcrumb trail shows 'Dashboard > Manage Jenkins'. The main content area has two sections: 'various nodes that Jenkins runs jobs on.' and 'to provision agents on-demand.' Below this, under 'Managed files', it says 'e.g. settings.xml for maven, central managed scripts, custom files, ...'. Under 'Security', there are three sections: 'Security' (Secure Jenkins; define who is allowed to access/use the system), 'Credentials' (Configure credentials), and 'Credential Providers' (Configure the credential providers and types). Further down, under 'Status Information', there are three sections: 'System Information' (Displays various environmental information to assist trouble-shooting), 'System Log' (System log captures output from java.util.logging output related to Jenkins), and 'Load Statistics' (Check your resource utilization and see if you need more computers for your builds). At the bottom of the page, a toolbar includes links for 'About Jenkins', 'Jenkins', 'Search', and 'log out'. The status bar at the bottom right shows the date '19-07-2024' and time '14:05'.

Click on system.

The screenshot shows the Jenkins Manage Jenkins interface, specifically the 'Credentials' section. The top navigation bar and search bar are identical to the previous screenshot. The breadcrumb trail shows 'Dashboard > Manage Jenkins > Credentials'. The main content area is titled 'Credentials' and includes a table header with columns 'T', 'P', 'Store ↓', 'Domain', 'ID', and 'Name'. Below the table, a section titled 'Stores scoped to Jenkins' shows a table with one row: 'System' (Icon: User) under 'Domains' (Value: '(global)'). At the bottom of the page, a toolbar includes links for 'About Jenkins', 'Jenkins', 'Search (CTRL+K)', and 'log out'. The status bar at the bottom right shows the date '19-07-2024' and time '14:06'.

Click on global credentials (unrestricted).

The screenshot shows the Jenkins 'System' credentials management interface. It lists a single credential named 'Global credentials (unrestricted)' with the description: 'Credentials that should be available irrespective of domain specification to requirements matching.' Below the table are icons for sorting by 'Domain' (downward arrow) and 'Description'. A blue button labeled '+ Add domain' is located at the top right. The browser's address bar shows the URL: 54.226.50.218:8080/manage/credentials/store/system/. The top navigation bar includes links for Instances | EC2, Stacks | CloudFormation, Clusters | Elastic, EC2 Instance Config, IAM, S3 buckets | S3, System [Jenkins], Pipeline Syntax, Users - Administer, and log out.

System

+ Add domain

Domain ↓

Description



Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

Icon: S M L

REST API Jenkins 2.452.3

The screenshot shows a Windows taskbar with several pinned icons, including File Explorer, Mail, and Edge. The system tray shows the date (19-07-2024), time (14:06), battery level, signal strength, and network status. The taskbar also includes a search bar and a language indicator (ENG IN).

Choose kind as secret text.

Paste the sonar token in secret and click on sonar-token.

Click on create.

The screenshot shows the 'New credentials' creation form. The 'Kind' dropdown is set to 'Secret text'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc.)'. The 'Secret' field contains a long string of asterisks ('*****'). The 'ID' field is filled with 'Sonar-token'. The 'Description' field is filled with 'Sonar-token'. At the bottom, a blue 'Create' button is visible. The browser's address bar shows the URL: 54.226.50.218:8080/manage/credentials/store/system/domain/_/newCredentials. The top navigation bar includes links for Instances | EC2, Stacks | CloudFormation, Clusters | Elastic, EC2 Instance Config, IAM, S3 buckets | S3, System [Jenkins], Pipeline Syntax, Users - Administer, and log out. The taskbar at the bottom shows the date (19-07-2024), time (14:07), battery level, signal strength, and network status.

New credentials

Kind

Secret text

Scope ?

Global (Jenkins, nodes, items, all child items, etc.)

Secret

ID ?

Sonar-token

Description ?

Sonar-token

Create

The screenshot shows a Windows taskbar with several pinned icons, including File Explorer, Mail, and Edge. The system tray shows the date (19-07-2024), time (14:07), battery level, signal strength, and network status. The taskbar also includes a search bar and a language indicator (ENG IN).

Global credentials (unrestricted)

ID	Name	Kind	Description
Sonar-token	Sonar-token	Secret text	Sonar-token

Icon: S M L



Now, go to Dashboard → Manage Jenkins → System.

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

Warnings have been published for the following currently installed components:

docker-build-step 2.12:
CSRF vulnerability and missing permission check (no fix available)
No fixes for these issues are available. It is recommended that you review the security advisory and apply mitigations if possible, or uninstall this plugin.

Go to plugin manager | Configure which of these warnings are shown

System Configuration

- System**: Configure global settings and paths.
- Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Tools**: Configure tools, their locations and automatic installers.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Appearance**: Configure the look and feel of Jenkins.

Give name and sonar server URL.

Select our token.

Click on apply and save

The screenshot shows the Jenkins System configuration page under the 'Manage Jenkins' section. The 'SonarQube servers' section is active. It contains fields for 'Name' (set to 'sonar-server'), 'Server URL' (set to 'http://54.226.50.218:9000'), and 'Server authentication token' (set to 'Sonar-token'). A 'Save' button is visible at the bottom.

Go to SonarQube server.

Administration → Configuration → Webhooks

The screenshot shows the SonarQube Administration page under the 'sonarqube' project. The 'Administration' tab is selected. In the 'Webhooks' section, there is a table with one row for 'Administrator admin'. The table columns are 'SCM Accounts', 'Last connection', 'Groups', and 'Tokens'. The 'Tokens' column shows '1' token. A 'Create User' button is located at the top right of the user list.

Click on create.

No webhook defined.

Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for it.

SonarQube™ technology is powered by SonarSource SA
Community Edition - v9.9.6 (build 92038) - [GPL v3](#) - [Community](#) - [Documentation](#)

Get the most out of SonarQube!

Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.

Learn More

Dismiss

Add details

Give name

#in url section of quality gate

<http://jenkins-public-ip:8080>/sonarqube-webhook/>

Create Webhook

All fields marked with * are required

Name *

URL *

Secret

If provided, secret will be used as the key to generate the HMAC hex (lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256' header.

Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for it.

SonarQube™ technology is powered by SonarSource SA
Community Edition - v9.9.6 (build 92038) - [GPL v3](#) - [Community](#) - [Documentation](#)

Get the most out of SonarQube!

Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.

Learn More

Dismiss

Now add Docker credentials to the Jenkins to log in and push the image

Manage Jenkins → Credentials → global → add credential

The screenshot shows the Jenkins Global credentials (unrestricted) page. It lists a single credential named "Sonar-token" with a secret text kind. The page includes a search bar, a "Add Credentials" button, and a toolbar with icons for S, M, and L.

ID	Name	Kind	Description
Sonar-token	Sonar-token	Secret text	Sonar-token

REST API Jenkins 2.452.3
54.226.50.218:8080/manage/credentials/store/system/domain/_/newCredentials
27°C Mostly cloudy 14:12 19-07-2024

Add Docker Hub Username and Password under Global Credentials

The screenshot shows the Jenkins New credential page for adding a "Username with password" credential. The form fields include:

- Kind: Username with password
- Scope: Global (Jenkins, nodes, items, all child items, etc)
- Username: chinni111
- Treat username as secret:
- Password: (redacted)
- ID: docker
- Description: docker

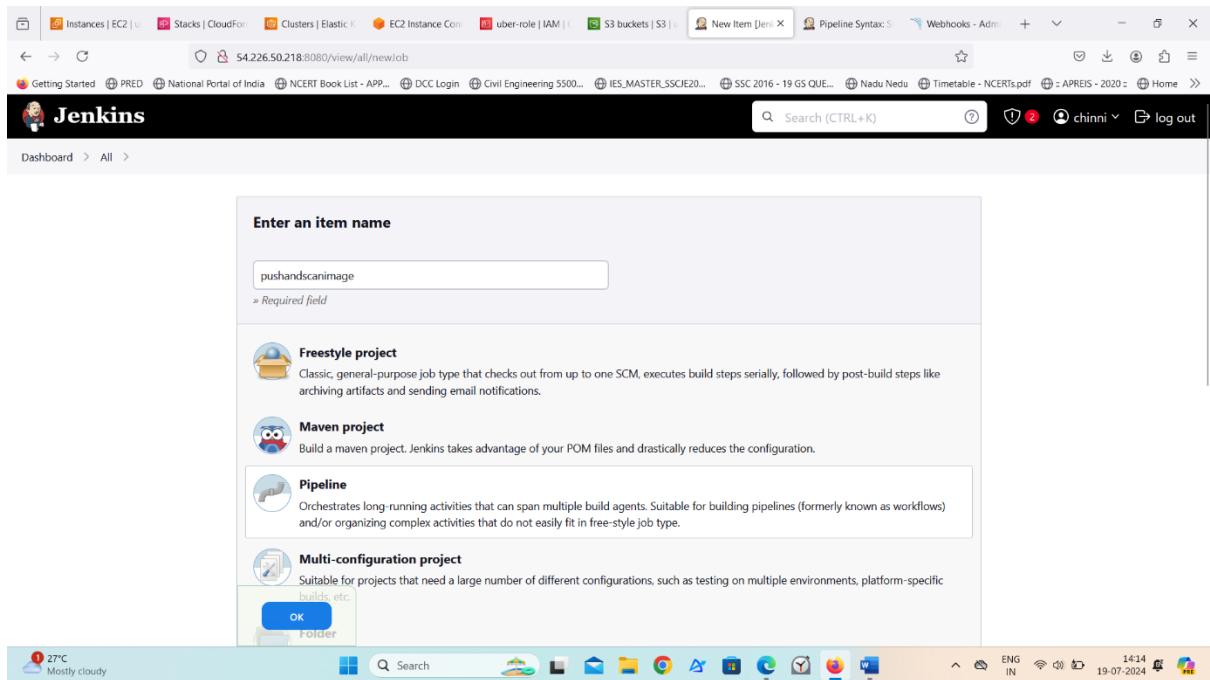
A "Create" button is at the bottom left. The page is part of the Jenkins interface with standard navigation and status bars at the top and bottom.

REST API Jenkins 2.452.3
54.226.50.218:8080/manage/credentials/store/system/domain/_/newCredentials
27°C Mostly cloudy 14:13 19-07-2024

Click on Create.

Step10: Pipeline up to Docker

Now let's create a new job for our pipeline



Add this to Pipeline

```
pipeline{
```

```
    agent any
```

```
    tools{
```

```
        jdk 'jdk17'
```

```
        nodejs 'node16'
```

```
}
```

```
    environment {
```

```
        SCANNER_HOME=tool 'sonar-scanner'
```

```
}
```

```
    stages {
```

```
        stage('clean workspace'){
```

```
            steps{
```

```
                cleanWs()
```

```
}
```

```
}
```

```
        stage('Checkout from Git'){
```

```

steps{
    checkout scmGit(branches: [[name: '/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/vommida-puchinni/uber-clone.git']])
}

stage("Sonarqube Analysis"){
    steps{
        withSonarQubeEnv('sonar-server') {
            sh "$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Uber \
-Dsonar.projectKey=Uber"
        }
    }
}

stage("quality gate"){
    steps {
        script {
            waitForQualityGate abortPipeline: false, credentialsId: 'Sonar-token'
        }
    }
}

stage('Install Dependencies') {
    steps {
        sh "npm install"
    }
}

stage('OWASP FS SCAN') {

```

```

steps {
    dependencyCheck additionalArguments: '--scan ./ --disableYarnAudit
--disableNodeAudit', odcInstallation: 'DP-Check'
    dependencyCheckPublisher pattern: '**/dependency-check-
report.xml'
}

stage('TRIVY FS SCAN') {
    steps {
        sh "trivy fs . > trivyfs.txt"
    }
}

stage("Docker Build & Push"){
    steps{
        script{
            withDockerRegistry(credentialsId: 'docker', toolName: 'docker'){
                sh "docker build -t uber ."
                sh "docker tag uber chinni111/uber:latest "
                sh "docker push chinni111/uber:latest "
            }
        }
    }
}

stage("TRIVY"){

    steps{
        sh "trivy image chinni111/uber:latest > trivyimage.txt"
    }
}

```

```

stage("deploy_docker"){
    steps{
        sh "docker run -d --name uber -p 3000:3000 chinni111/uber:latest"
    }
}
}
}
}

```

Screenshot of the AWS CodePipeline configuration interface showing the Pipeline script editor.

```

Configure Pipeline Definition Pipeline script

Script ? 1 * pipeline{ 2     agent any 3     tools{ 4         nodejs 'Node17' 5         nodejs 'Node10' 6     } environment{ 7         SCANNER_HOME_tool 'sonar-scanner' 8     } stages{ 9         stage('clean workspace'){ 10             steps{ 11                 clean() 12             } 13         } 14         stage('Checkout from Git'){ 15             steps{ 16                 checkout scm@{branches: [[name: '/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/vommidaupuchinni/uber']]} 17             } 18         } 19         stage('SonarQube Analysis'){ 20             steps{ 21                 withSonarScanner('sonar-server'){ 22                     sh "$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Uber" 23                 } 24             } 25         } 26     } 27     stage('quality_gate'){ 28         steps{ 29             script{ 30                 waitforQualityGate abortPipeline: false, credentialsId: 'Sonar-token' 31             } 32         } 33     } 34 } 35 } 36 } 37 } 38 } 39 } 40 } 41 } 42 } 43 } 44 } 45 } 46 } 47 } 48 } 49 } 50 } 51 } 52 } 53 } 54 } 55 } 56 } 57 } 58 } 59 } 60 } 61 } 62 } 63 } 64 } 65 } 66 } 67 } 68 } 69 } 70 } 71 } 72 } 73 } 74 } 75 }

Save Apply

```

Operating System Taskbar:

Screenshot of the AWS CodePipeline configuration interface showing the Pipeline script editor.

```

Configure Pipeline Definition Pipeline script

Script ? 1 * pipeline{ 2     agent any 3     tools{ 4         nodejs 'Node17' 5         nodejs 'Node10' 6     } environment{ 7         SCANNER_HOME_tool 'sonar-scanner' 8     } stages{ 9         stage('Install Dependencies'){ 10             steps{ 11                 sh 'npm install' 12             } 13         } 14         stage('OWASP FS SCAN'){ 15             steps{ 16                 dependencyCheck additionalArguments: '--scan ./ --disableVarnAudit --disableNodeAudit', odcInstallation: 'DP-Check' 17                 dependencyPublisher pattern: '**/dependency-check-report.xml' 18             } 19         } 20         stage('TRIVY FS SCAN'){ 21             steps{ 22                 sh 'trivy fs . > trivyfs.txt' 23             } 24         } 25         stage('Docker Build & Push'){ 26             steps{ 27                 script{ 28                     withDockerRegistry(credentialsId: 'docker', toolName: 'docker'){ 29                         sh 'docker build -t uber .' 30                         sh 'docker tag uber chinni111/uber:latest' 31                         sh 'docker push chinni111/uber:latest' 32                     } 33                 } 34             } 35         } 36         stage('TRIVY'){ 37             steps{ 38                 sh 'trivy image chinni111/uber:latest > trivymain.txt' 39             } 40         } 41         stage('deploy_docker'){ 42             steps{ 43                 sh 'docker run -d --name uber -p 3000:3000 chinni111/uber:latest' 44             } 45         } 46     } 47 } 48 } 49 } 50 } 51 } 52 } 53 } 54 } 55 } 56 } 57 } 58 } 59 } 60 } 61 } 62 } 63 } 64 } 65 } 66 } 67 } 68 } 69 } 70 } 71 } 72 } 73 } 74 } 75 }

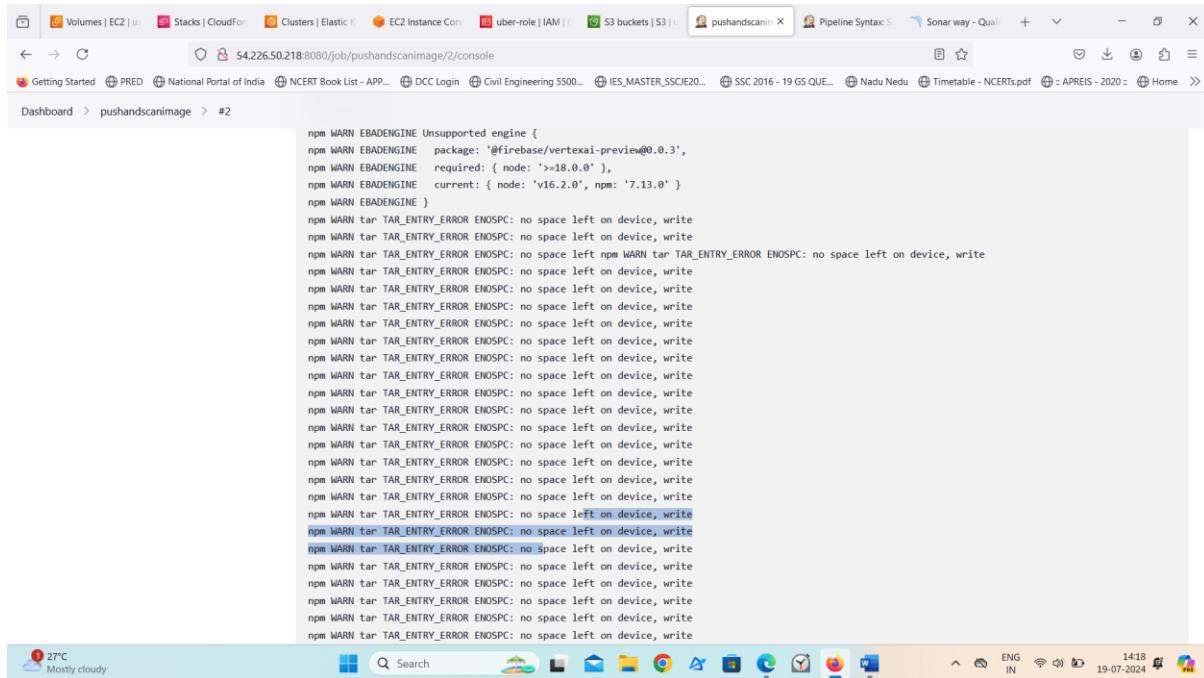
Save Apply

```

Operating System Taskbar:

Click on apply and then save it.

Click on build

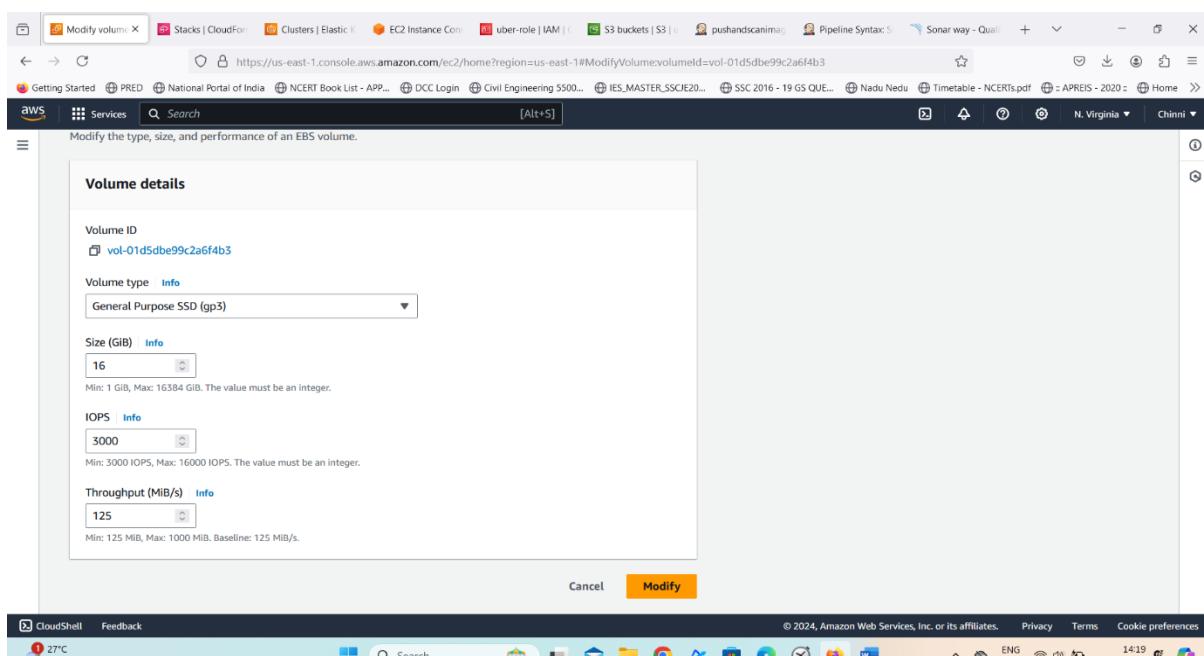


The screenshot shows a browser window with multiple tabs open. The active tab displays a series of identical npm warning messages:

```
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: '@firebase/vertexai-preview@0.0.3',
npm WARN EBADENGINE   required: { node: '>=18.0.0' },
npm WARN EBADENGINE   current: { node: 'v16.2.0', npm: '7.13.0' }
npm WARN EBADENGINE }
```

Below the terminal output, the system tray shows the date and time as 19-07-2024 14:18, along with other icons like battery level and signal strength.

- It showing no space left on my device.
- To increase the storage capacity. Go to volume attached to our ec2 instance.
- Click on volume action.
- Choose modify volume and increase the size of the volume.
- Click on modify.



The screenshot shows the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyVolume?volumeld=vol-01d5dbe99c2a6f4b3>. The page is titled "Modify volume" and displays "Volume details".

Volume ID: vol-01d5dbe99c2a6f4b3

Volume type: General Purpose SSD (gp3)

Size (GiB): 16

IOPS: 3000

Throughput (MiB/s): 125

At the bottom, there are "Cancel" and "Modify" buttons, with "Modify" being highlighted.

Now go to instance click on instance state, click on reboot instance.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main area displays two instances: 'i-038943490d4eb2102' and 'jenkins-server'. The 'jenkins-server' instance is selected. A modal window titled 'i-082e4110e988f7b6e (jenkins-server)' provides detailed information about the instance, including its Public IPv4 address (54.226.50.218) and Private IPv4 addresses (172.31.19.92). At the top of the main table, there's an 'Actions' dropdown menu with several options: Stop instance, Start instance, Reboot instance, Hibernate instance, and Terminate instance. The 'Reboot instance' option is currently highlighted.

Login to Jenkins.

The screenshot shows the Jenkins sign-in page. On the left, there's a large, colorful background image featuring a cartoon character of a man in a suit and bow tie. To the right of the image, the title 'Sign in to Jenkins' is displayed. Below the title is a form with three fields: 'Username' (with a placeholder 'johndoe'), 'Password' (with a placeholder 'password'), and a 'Keep me signed in' checkbox. At the bottom of the form is a prominent blue 'Sign in' button. The URL in the browser's address bar is 54.226.50.218:8080/login?from=%2Fjob%2Fterraform-eks%2Fpipeline-syntax%2F.

Start sonar server.

Using docker start sonar

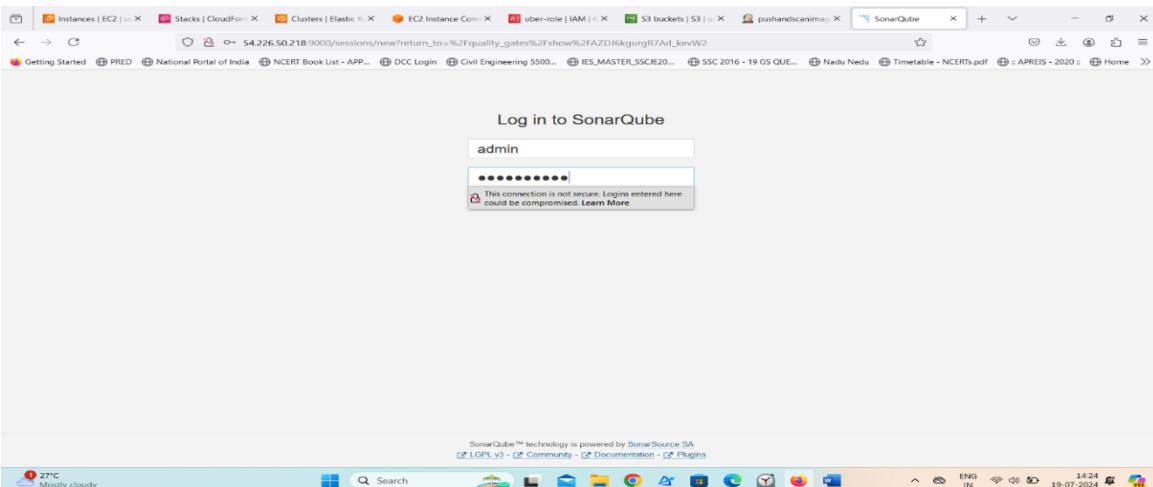
```

Instances | EC2 | u x Stacks | CloudFormation x Clusters | Elastic x EC2 Instance Conn x uber-role | IAM x S3 buckets | S3 x pushandsanimag x SonarQube x + - _ o x
Getting Started PRED National Portal of India NCERT Book List - APP... DCC Login Civil Engineering 5500... IES_MASTER_SSCE20... SSC 2016 - 19 GS QUE... Nadu Nedu Timetable - NCERTs.pdf APREIS - 2020 Home >
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-082e4110e988f7b6e&osUser=ubuntu&region=us-east-1
aws Services Search [Alt+S]
System load: 0.45 Processes: 181
Usage of /: 44.5% of 14.46GB Users logged in: 0
Memory usage: 12% IPv4 address for enx0: 172.31.19.92
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
27 updates can be applied immediately.
13 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

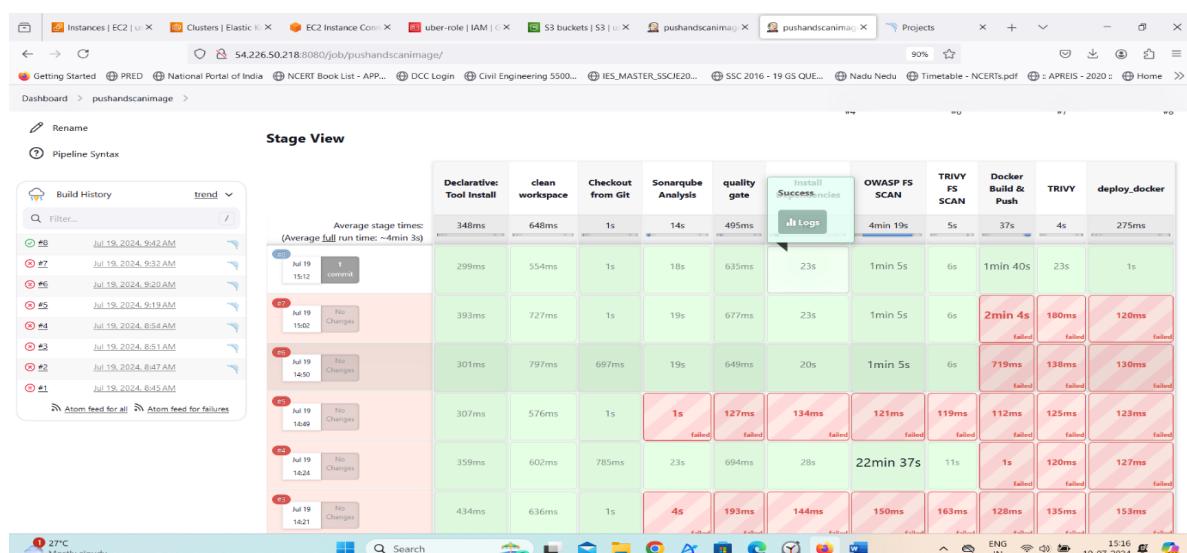
last login: Fri Jul 19 06:50:09 2024 from 10.206.107.27
ubuntu@ip-172-31-19-92:~$ sudo -l
root@ip-172-31-19-92:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
root@ip-172-31-19-92:~# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
root@ip-172-31-19-92:~# /opt/sonarqube/docker/run.sh
root@ip-172-31-19-92:~# docker start sonar
sonar
root@ip-172-31-19-92:~# i-082e4110e988f7b6e (jenkins-server)
PublicIPs: 54.226.50.218 PrivateIPs: 172.31.19.92

```

Login to SonarQube server.



Go to Jenkins and click on our pipeline. Click on build.



We see our images created.

```

root@ip-172-31-19-92:~# docker start sonar
sonar
root@ip-172-31-19-92:~# sudo usermod -aG docker jenkins
root@ip-172-31-19-92:~# sudo systemctl restart docker
root@ip-172-31-19-92:~# groups jenkins
jenkins : jenkins docker
root@ip-172-31-19-92:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND           CREATED          STATUS          PORTS     NAMES
cf78e63e03b   sonarqube:its-community   "/opt/sonarqube/docker..."   2 hours ago    Exited (137)   41 seconds ago
root@ip-172-31-19-92:~# docker start sonar
sonar
root@ip-172-31-19-92:~# sudo systemctl restart jenkins
root@ip-172-31-19-92:~# docker images
REPOSITORY      TAG          IMAGE ID            CREATED          SIZE
devopsvmr/uber  latest       14b7798acd75   About a minute ago   1.2GB
uber            latest       f4b7798acd75   About a minute ago   1.2GB
node            lts-alpine   3ef52edada8a70   10 days ago    13.9MB
sonarqube       lts-community 9194edad30a2   3 weeks ago    603MB
root@ip-172-31-19-92:~# docker rmi devopsvmr/uber:latest -f
Untagged: devopsvmr/uber:latest
i-082e4110e988f7b6e (jenkins-server)
PublicIPs: 54.226.50.218 PrivateIPs: 172.31.19.92

```

```

root@ip-172-31-19-92:~# docker images
REPOSITORY      TAG          IMAGE ID            CREATED          SIZE
chininilli/uber  latest       6824b23fb067   3 minutes ago   1.2GB
ubuntu          latest       6824b23fb067   3 minutes ago   1.2GB
node            lts-alpine   3ef52edada8a70   10 days ago    13.9MB
sonarqube       lts-community 9194edad30a2   3 weeks ago    603MB
root@ip-172-31-19-92:~#

```



Now go to SonarQube server.

You can see the report has been generated and the status shows as passed. You can see that there are 641 lines it scanned. To see a detailed report, you can go to issues.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

My Favorites All

Search for projects. A

1 project(s)

Perspective: Overall Status Sort by: Name

Uber Passed

Last analysis: 7 minutes ago

Bugs: 0 A Vulnerabilities: 1 E Hotspots Reviewed: 0.0% Code Smells: 3 A Coverage: 0.0% Duplications: 0.0% Lines: 641 JavaScript

Filters

Quality Gate: Passed (1) Failed (0)

Reliability (Bug Rating): A rating (1) B rating (0) C rating (0) D rating (0) E rating (0)

Security (Vulnerabilities): A rating (0) B rating (0) C rating (0) D rating (0) E rating (1)

Security Review (Security Hotspots): ≥ 80% (0) 70% - 80% (0) 50% - 70% (0)

Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions, and it will not support switching to a different database engine.

Get the most out of SonarQube! SonarQube™ technology is powered by SonarSource. Community Edition - v9.0.6 (build 92038) - LGPL v3 - Community - Documentation

Light rain At night

ENG IN 19-07-2024

Click on quality gates we see the results and the created quality gate.

Sonar way - Qual X + - ⌂

54.226.50.218:9000/quality_gates/show/AZDj6kgurgR7Ad_kevW2

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Sonar way BUILT-IN

Sonar way DEFAULT BUILT-IN

This quality gate complies with Clean as You Code

This quality gate complies with the Clean as You Code methodology, so that you benefit from the most efficient approach to delivering Clean Code. It ensures that:

- No new bugs are introduced
- No new vulnerabilities are introduced
- All new security hotspots are reviewed
- New code has limited technical debt
- New code has limited duplication
- New code is properly covered by tests

Conditions

Conditions on New Code

Metric	Operator	Value
Coverage	is less than	80.0%
Duplicated Lines (%)	is greater than	3.0%
Maintainability Rating	is worse than	A (Technical debt ratio is less than 5.0%)
Reliability Rating	is worse than	A (No bugs)
Security Hotspots Reviewed	is less than	100%
Security Rating	is worse than	A (No vulnerabilities)

Projects

Light rain At night

ENG IN 19-07-2024

The screenshot shows a Jenkins pipeline console output. The pipeline starts by cloning a repository from GitHub ('Started by user chinni'), then it runs a series of stages: 'node' (using node.js), 'Running on Jenkins in /var/lib/jenkins/workspace/pushandscanimage', 'stage' (with declarative tool install), 'stage' (with env vars for tool), 'stage' (clean workspace), 'stage' (with env vars for tool), 'stage' (with env vars for tool), 'stage' (with env vars for tool), and finally 'stage' (clean workspace). The pipeline concludes with '[WS-CLEANUP] Deleting project workspace...', '[WS-CLEANUP] Deferred wipeout is used...', and '[WS-CLEANUP] done'. The Jenkins interface includes a sidebar with various management options like Status, Changes, Pipeline Overview, and Pipeline Steps.

```
Instances | EC2 | Clusters | Elastic K... | EC2 Instance Con... | uber-role | IAM | S3 buckets | S3 | pushandscanimage | pushandscanimage | Projects
Getting Started | PRED | National Portal of India | NCERT Book List - APP... | DCC Login | Civil Engineering 5500... | IES_MASTER_SSCE20... | SSC 2016 - 19 GS QUE... | Nadu Nedu | Timetable - NCERFs.pdf | APREIS - 2020 | Home > Dashboard > pushandscanimage > #8

[Pipeline] [
[Pipeline] dependencyCheck
[INFO] Checking for updates
[INFO] Skipping the NVD API Update as it was completed within the last 240 minutes
[INFO] Skipping Known Exploited Vulnerabilities update check since last check was within 24 hours.
[INFO] Check for updates complete (451 ms)
[INFO]

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

About ODC: https://jeremylong.github.io/DependencyCheck/general/internals.html
False Positives: https://jeremylong.github.io/DependencyCheck/general/suppressions.html

Sponsor: https://github.com/sponsors/jeremylong

[INFO] Analysis Started
[INFO] Finished File Name Analyzer (0 seconds)
[WARN] dependency skipped: package.json contain an alias for string-width-cjs > string-width@^4.2.0 npm audit doesn't support aliases
[WARN] dependency skipped: package.json contain an alias for strip-ansi-cjs > strip-ansi@^6.0.1 npm audit doesn't support aliases
[WARN] dependency skipped: package.json contain an alias for wrap-ansi-cjs > wrap-ansi@^7.0.0 npm audit doesn't support aliases
[WARN] dependency skipped: node module @next/swc-android-arm64 seems optional and not installed
[WARN] dependency skipped: node module @next/swc-android-arm64 seems optional and not installed
[WARN] dependency skipped: node module @next/swc-linux-arm64 seems optional and not installed
[WARN] dependency skipped: node module @next/swc-linux-arm64-gpu seems optional and not installed
[WARN] dependency skipped: node module @next/swc-linux-arm64-musl seems optional and not installed
[WARN] dependency skipped: node module @next/swc-win32-arm64-msvc seems optional and not installed
[WARN] dependency skipped: node module @next/swc-win32-i386-msvc seems optional and not installed
[WARN] dependency skipped: node module @next/swc-win32-x64 seems optional and not installed
```

```
Getting Started PRED National Portal of India NCERT Book List - APP... DCC Login Civil Engineering 5500... IES_MASTER_SSCEIE20... SSC 2016 - 19 GS QUE... Nadu Nedu Timetable - NCERTs.pdf = APREIS - 2020 Home > # Instances | EC2 | Clusters | Elastic K. | EC2 Instance Con... | uber-role | IAM | S3 buckets | S3 | pushandscanimage | pushandscanimage | Projects

54.226.50.218:8080/job/pushandscanimage/8/console

[Pipeline] {
  [Pipeline] script
  [Pipeline] {
    [Pipeline] withDockerRegistry
      $ /var/lib/jenkins/tools/org.jenkinsci.plugins.docker.commons.tools.DockerTool/docker/bin/docker login -u chinmili -p ***** https://index.docker.io/v1/
    Login Succeeded
    [Pipeline] {
      [Pipeline] sh
      + docker build -t uber .
      DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
      Install the buildx component to build images with BuildKit:
      https://docs.docker.com/go/buildx/
      
      Sending build context to Docker daemon 481.4MB

      Step 1/8 : FROM node:alpine
      --> 3ef52edaa70
      Step 2/8 : WORKDIR /app
      --> Using cache
      --> 25d25a0a05e8
      Step 3/8 : COPY package.json package-lock.json /app/
      --> Using cache
      --> befeec3d7722
      Step 4/8 : RUN npm install
      --> Using cache
      --> 53156347c9d9
      Step 5/8 : COPY . /app/
      --> d192158671ae
      Step 6/8 : RUN npm run build
      --> Running in 3fcdf709739

      > build
      > next build

  }
}

```

```

Instances | EC2 | X Clusters | Elastic | X EC2 Instance Com X uber-role | IAM | X S3 buckets | S3 | X pushandsanimage X Projects X + - ? X
Getting Started PRED National Portal of India NCERT Book List - APP... DCC Login Civil Engineering 5500... IES,MASTER_SSCE20... SSC 2016 - 19 GS QUE... Nadu Nedu Timetable - NCERTs.pdf APREIS - 2020 Home >
Dashboard > pushandsanimage > #8
Successfully tagged uber:latest
[Pipeline] sh
[Pipeline] $ docker tag uber chinmili111/uber:latest
[Pipeline]
+ docker push chinmili111/uber:latest
The push refers to repository [docker.io/chinmili111/uber]
ae116609bf9c: Preparing
418c36ceef32: Preparing
418c36ceef33: Preparing
d153ef177852: Preparing
6a9f92f44e79: Preparing
ea024285a3e0: Preparing
87ea97051ac: Preparing
9e61ad6794e0: Preparing
9d45f6ff6f8e3: Preparing
ea024285a3e0: Waiting
87ea97051ac: Waiting
0c13ad6794e0: Waiting
94e5f6ff6f8e3: Waiting
94e5f6ff6f8e3: Pushed
d153ef177852: Pushed
ea024285a3e0: Mounted from library/node
87ea97051ac: Mounted from library/node
0c13ad6794e0: Mounted from library/node
94e5f6ff6f8e3: Mounted from library/node
ae116609bf9c: Pushed
418c36ceef32: Pushed
4333fe3c2e7a3: Pushed
latest: digest: sha256:14af61e46f0ed00fd94af02f4197c5dfb540c81cd05ab6c4df423f0c21101c0 size: 2211
[Pipeline]
[Pipeline] // withDockerRegistry
[Pipeline]
[Pipeline] // script
10Locs, 3 s

```

```

Instances | EC2 | X Clusters | Elastic | X EC2 Instance Com X uber-role | IAM | X S3 buckets | S3 | X pushandsanimage X pushandsanimage X Projects X + - ? X
Getting Started PRED National Portal of India NCERT Book List - APP... DCC Login Civil Engineering 5500... IES,MASTER_SSCE20... SSC 2016 - 19 GS QUE... Nadu Nedu Timetable - NCERTs.pdf APREIS - 2020 Home >
Dashboard > pushandsanimage > #8
2024-07-19T09:46:18Z INFO [alpine] Detecting vulnerabilities... os_version="3.20" repository="3.20" pkg_num=16
2024-07-19T09:46:18Z INFO Number of language-specific files num=1
2024-07-19T09:46:18Z INFO [node-pkg] Detecting vulnerabilities...
2024-07-19T09:46:18Z WARNING Using package files from other vendors for some vulnerabilities. Read https://aquasecurity.github.io/trivy/v0.53/docs/scanner/vulnerability-key-selectivity-selection.html for details.
2024-07-19T09:46:18Z INFO Table result includes only package filenames. Use '--format json' option to get the full path to the package file.
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // tool
[Pipeline] [deploy_docker]
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline]
[Pipeline] [
[Pipeline] docker run d --name uber -p 3000:3000 chinmili111/uber:latest
2670100d7d9e36ca8aa1eeb93c0e42029b31cab78f25db287f71d3da8c06f612
[Pipeline]
[Pipeline] // withEnv
[Pipeline] [
[Pipeline] // stage
[Pipeline]
[Pipeline] [
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline] end of Pipeline
Finished: SUCCESS

```

OWASP, you will see that in status, a graph will also be generated and Vulnerabilities.

Dependency-Check Results				
Severity Distribution				
	File Name	Vulnerability	Severity	Weakness
+	json5:1.0.1	NVD CVE-2022-46175	High	CWE-1321
+	micromatch:4.0.7	OSSINDEX CVE-2024-4067	Medium	CWE-1333
+	minimist:1.2.5	NVD CVE-2021-44906	Critical	CWE-1321
+	packages-bundle.js	NVD CVE-2021-23337	High	CWE-94
+	packages-bundle.js	NVD CVE-2020-28500	Medium	NVD-CWE-Other
+	postcss:8.4.5	NVD CVE-2023-44270	Medium	CWE-74
+	semver:6.3.0	OSSINDEX CVE-2022-25883	High	CWE-1333
+	semver:7.3.5	OSSINDEX CVE-2022-25883	High	CWE-1333
+	ua-parser.js	NVD CVE-2022-25927	High	CWE-1333
+	undici:5.28.4	OSSINDEX CVE-2024-24750	Medium	CWE-400

Dependency-Check Results

SEVERITY DISTRIBUTION

File Name	Vulnerability	Severity	Weakness
word-wrap:1.2.3	NVD CVE-2023-26115	High	CWE-1333

When you log in to Docker hub, you will see a new image is created

docker hub Explore Repositories Organizations

Search Docker Hub

chinni111 / **uber**
Contains: Image • Last pushed: 5 minutes ago

chinni111 / **tomcat**
Contains: Image • Last pushed: 21 days ago

chinni111 / **myapp**
Contains: Image • Last pushed: 22 days ago

chinni111 / **chinni**
Contains: No content • Created: 23 days ago

Create An Organization

Create and manage users and grant access to your repositories.

```
root@ip-172-31-19-92:~# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
chinni111/uber latest 6824b23fb067 3 minutes ago 1.26B
uber latest 6824b23fb067 3 minutes ago 1.26B
node latest 3845290a70 10 days ago 133M
sonarcube lts-community 9194aded30a2 3 weeks ago 603MB
root@ip-172-31-19-92:~# trivy img: chinni111/uber:latest
2024-07-19T09:50:03Z INFO Need to update DB
2024-07-19T09:50:03Z INFO Downloading DB... repository="ghcr.io/aquasecurity/trivy-db:2"
2024-07-19T09:50:06Z INFO Vulnerability scanning is enabled
2024-07-19T09:50:06Z INFO Secret scanning is enabled
2024-07-19T09:50:06Z INFO If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2024-07-19T09:50:06Z INFO Please see also https://aquasecurity.github.io/trivy/v0.53/docs/scanner/secret#recommendation-for-faster-secret-detection
2024-07-19T09:50:06Z INFO Detecting vulnerabilities in image chinni111/uber:latest
2024-07-19T09:50:28Z INFO [alpine] detecting vulnerabilities... at version="3.20" repository="3.20" pkg_num=1
2024-07-19T09:50:28Z INFO Number of language-specific files num=1
2024-07-19T09:50:28Z INFO [node-pkg] detecting vulnerabilities...
2024-07-19T09:50:28Z WARN Using severities from other vendors for some vulnerabilities. Read https://aquasecurity.github.io/trivy/v0.53/docs/scanner/vulnerability_severity_selection_for_details.
chinni111/uber:latest (alpine 3.20.1)

total: 2 (UNKNOWN: 0, LOW: 0, MEDIUM: 2, HIGH: 0, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
i-082e4110e988f7b6e (jenkins-server)						

Using trivy image imagename: tag we can see our severity and vulnerability in our command line tool (means in server).

```

2024-07-19T09:50:28Z INFO Table result includes only package filenames. Use '--format json' option to get the full path to the package file.
Node _js (node-pkg)
Total: 7 (UNKNOWN: 0, LOW: 1, MEDIUM: 4, HIGH: 1, CRITICAL: 1)

| Library | Vulnerability | Severity | Status | Installed Version | Fixed Version | Title |
| --- | --- | --- | --- | --- | --- | --- |
| openssl | CVE-2024-5535 | MEDIUM | fixed | 3.3.1-r0 | 3.3.1-r1 | openssl: SSL select_next_proto buffer overread https://avd.aquasec.com/nvd/cve-2024-5535 |
| libssl3 |  |  |  |  |  |  |

2024-07-19T09:50:28Z INFO Table result includes only package filenames. Use '--format json' option to get the full path to the package file.
Node _js (node-pkg)
Total: 7 (UNKNOWN: 0, LOW: 1, MEDIUM: 4, HIGH: 1, CRITICAL: 1)

| Library | Vulnerability | Severity | Status | Installed Version | Fixed Version | Title |
| --- | --- | --- | --- | --- | --- | --- |
| json5 (package.json) | CVE-2022-46175 | HIGH | fixed | 1.0.1 | 2.2.2, 1.0.2 | json5: Prototype Pollution in JSON via Parse Method https://avd.aquasec.com/nvd/cve-2022-46175 |
|  |  |  |  |  |  |  |

i-082e4110e988f7b6e (jenkins-server)
Public IPs: 54.226.50.218 Private IPs: 172.31.19.92
  
```



```

2024-07-19T09:50:28Z INFO Table result includes only package filenames. Use '--format json' option to get the full path to the package file.
Node _js (nodejs-semver)
Total: 7 (UNKNOWN: 0, LOW: 1, MEDIUM: 4, HIGH: 1, CRITICAL: 1)

| Library | Vulnerability | Severity | Status | Installed Version | Fixed Version | Title |
| --- | --- | --- | --- | --- | --- | --- |
| minimist (package.json) | CVE-2021-44906 | CRITICAL | | 1.2.5 | 1.2.6, 0.2.4 | minimist: prototype pollution https://avd.aquasec.com/nvd/cve-2021-44906 |
| next (package.json) | CVE-2023-46298 | LOW | | 12.1.0 | 13.4.20-canary.13 | Next.js missing cache-control header may lead to CDN caching empty reply https://avd.aquasec.com/nvd/cve-2023-46298 |
| postcss (package.json) | CVE-2023-44270 | MEDIUM | | 8.4.5 | 8.4.31 | An issue was discovered in PostCSS before 8.4.31. The vulnerability affects ..... https://avd.aquasec.com/nvd/cve-2023-44270 |
| server (package.json) | CVE-2022-25883 |  | | 6.3.0 | 7.5.2, 6.3.1, 5.7.2 | nodejs-semver: Regular expression denial of service https://avd.aquasec.com/nvd/cve-2022-25883 |

i-082e4110e988f7b6e (jenkins-server)
Public IPs: 54.226.50.218 Private IPs: 172.31.19.92
  
```



```

2024-07-19T09:50:28Z INFO Table result includes only package filenames. Use '--format json' option to get the full path to the package file.
Node _js (word-wrap)
Total: 4 (UNKNOWN: 0, LOW: 1, MEDIUM: 3, HIGH: 0, CRITICAL: 0)

| Library | Vulnerability | Severity | Status | Installed Version | Fixed Version | Title |
| --- | --- | --- | --- | --- | --- | --- |
| postcss (package.json) | CVE-2023-44270 | MEDIUM | | 0.4.5 | 0.4.31 | An issue was discovered in PostCSS before 0.4.31. The vulnerability affects ..... https://avd.aquasec.com/nvd/cve-2023-44270 |
| server (package.json) | CVE-2022-25883 |  | | 6.3.0 | 7.5.2, 6.3.1, 5.7.2 | nodejs-semver: Regular expression denial of service https://avd.aquasec.com/nvd/cve-2022-25883 |
|  |  |  | | 7.3.5 |  |  |
| word-wrap (package.json) | CVE-2023-26115 |  | | 1.2.3 | 1.2.4 | word-wrap: ReDoS https://avd.aquasec.com/nvd/cve-2023-26115 |

root@ip-172-31-19-92:~#
i-082e4110e988f7b6e (jenkins-server)
Public IPs: 54.226.50.218 Private IPs: 172.31.19.92
  
```



- Using docker scout also we can scan our image.
- Click our pushed repo and go to its settings. Choose docker scout image analysis, click on save

chinni111 / Repositories / uber / General

Docker commands

To push a new tag to this repository:

```
docker push chinni111/uber:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest	Image		10 minutes ago	10 minutes ago

[See all](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more](#) about automated builds.

[Upgrade](#)

chinni111 / Repositories / uber / Settings

Image security insight settings

Features and controls that help you uncover, understand, and fix issues with your container images in Docker Hub

Docker Scout image analysis NEW

Image analysis is provided by Docker Scout. [Learn more](#) and [upgrade](#).
This account is on the Docker Scout Free tier. Upgrade for increased image analysis limits and additional software supply chain features.

Docker Scout image analysis NEW
Know when new CVEs impact your images, learn where they're introduced, and get recommendations for remediation options. [Enable repos in bulk](#) on Scout Dashboard.

Static scanning
Images will be scanned once when pushed and the vulnerability report saved at that point in time.

None

[Cancel](#) [Save](#)

Visibility settings ?
This repository is public.

Click on apps symbol on left side of our profile icon.

There we see docker scout click on it skip it for four times.

Now go to settings → repo settings choose our image.

Repository settings

Enable or disable image analysis for your repositories.

Using 1 out of 3 repositories from Docker Scout Free plan. [Upgrade](#)

Repository	Status	Scout Image analysis	Created at	Host name	Description
chinni	Empty	Inactive	23 days ago	hub.docker.com	
myapp	Not empty	Inactive	22 days ago	hub.docker.com	
tomcat	Not empty	Inactive	22 days ago	hub.docker.com	
uber	Not empty	Active	14 minutes ago	hub.docker.com	

Rows per page: 25 1-4 of 4

Click on images is left side dashboard. We like below.

Images

Lists all the repositories and images in organization chinni111.

Repository	Most Recent Image	OS/Arch	Last pushed	Vulnerabilities	Policies status
chinni111/uber hub.docker.com	latest	amd64	1 minute ago	2 Critical, 1 High, 3 Medium, 1 Low	2/6 Compliance, 0 Improved, 0 Worsened

Rows per page: 15 1-1 of 1

Click on vulnerabilities we see all our image vulnerabilities.

Vulnerabilities

An in-depth look at the vulnerabilities affecting your images.

Severity	Vulnerability	Package	CVSS score	Detected in	Fix available	Fix Version
Critical	CVE-2021-44906	pkg:npm/minimist@1.2.5	9.8	1 Image	Yes	1.2.6
Critical	CVE-2024-5535	pkg:apk/alpine/openssl@3.3.1-r0?os_name=alpine&...	9.1	1 Image	Yes	3.3.1-r1
High	CVE-2022-46175	pkg:npm/json5@1.0.1	7.1	1 Image	Yes	1.0.2
Medium	CVE-2022-25883	pkg:npm/semver@6.3.0	5.3	1 Image	Yes	6.3.1
Medium	CVE-2023-26115	pkg:npm/word-wrap@1.2.3	5.3	1 Image	Yes	1.2.4
Medium	CVE-2023-44270	pkg:npm/postcss@8.4.5	5.3	1 Image	Yes	8.4.31
Low	CVE-2023-46298	pkg:npm/next@12.1.0		1 Image	Yes	13.4.20-canary.13

Rows per page: 25 1-7 of 7

Total overview clicks on overview.

The screenshot shows the Docker Scout Overview page for the organization chinni111. The left sidebar includes links for Overview, Policies, Images, Base Images, Packages, Vulnerabilities, Exceptions, Integrations, Settings, Repository settings, Notifications (Beta), and Billing. The main content area displays four cards under the heading "DOCKER SCOUT POLICY":

- Default non-root user:** 0% (0/1 images comply). Violation: 1 (+1 in last 7 days).
- No copyleft licenses:** 100% (1/1 images comply). Packages: 0 (no change in last 7 days).
- No fixable critical or high vulnerabilities:** 0% (0/1 images comply). Vulnerabilities: 2 (+2 in last 7 days).
- No high-profile vulnerabilities:** 100% (1/1 images comply). Vulnerabilities: 0 (no change in last 7 days).

Come down we see vulnerabilities trends

The screenshot shows the Docker Scout policy details for the "No outdated base images" configuration. It includes a card for "Supply chain attestations" (0% compliance) and a chart titled "VULNERABILITIES Vulnerabilities trends". The chart shows a sharp increase in vulnerabilities starting around July 2nd, peaking at approximately 45 Critical and High severity vulnerabilities, before dropping sharply by July 14th. A legend indicates the scale from Low (blue) to Critical (red).

The screenshot shows the Docker Scout Overview page again, focusing on the "VULNERABILITIES Vulnerabilities trends" section. The chart highlights a specific data point for July 2nd, showing 48 vulnerabilities across three categories: Low (48), Medium (1), and Critical (0). Below the chart, a table provides detailed information for a recently disclosed vulnerability:

Severity	Vulnerability	CVSS Score	Disclosed	Images Impacted
Critical	CVE-2024-5535	9.1	20 days ago	1

Step11: Kubernetes Deployment

- Go to Putty of your Jenkins instance SSH and enter the below command
 - aws eks update-kubeconfig --name <CLUSTER NAME> --region <CLUSTER REGION>
 - aws eks update-kubeconfig --name EKS_CLOUD --region ap-south-1
 - Let's see the nodes
 - kubectl get nodes
 - Copy the config file to Jenkins master or the local file manager and save it
 - copy it and save it in documents or another folder save it as secret-file.txt
 - Note: create a secret-file.txt in your file explorer save the config in it and use this at the kubernetes credential section.
 - Install Kubernetes Plugin, once it's installed successfully

Dashboard > Manage Jenkins > Plugins

Plugins

Available plugins

Search (CTRL+K)

Install

Released

Kubernetes 4253.v7700d91739e5

Kubernetes Client API 6.10.0-240.v57880ce8b_0b_2

Kubernetes Credentials 174.va_36e093562d9

Kubernetes CLI 1.12.1

Kubernetes Credentials Provider 1.262.v2670ef7ea_0c5

Kubernetes : Pipeline :: DevOps Steps 1.6

go to manage Jenkins → manage credentials → Click on Jenkins global → add credentials.

Choose kind as secret file. browse the saved file.

Click on create.

New credentials

Kind

Secret file

Scope

Global (Jenkins, nodes, items, all child items, etc)

File

Browse... secret-file.txt

ID

k8s

Description

k8s

Create

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
Sonar-token	Sonar-token	Secret text	Sonar-token
docker	chinni111/***** (docker)	Username with password	docker
k8s	secret-file.txt (k8s)	Secret file	k8s

Icon: S M L

REST API Jenkins 2.452.3

final step to deploy on the Kubernetes cluster

I deployed in terraform pipeline.

```
stage('Deploy to kubernets'){
```

```
    steps{
```

```
        script{
```

```
            dir('K8S') {
```

```
                withKubeConfig(caCertificate: "", clusterName: "", contextName: "", credentialsId: 'k8s', namespace: "", restrictKubeConfigAccess: false, serverUrl: "") {
```

```
                    sh 'kubectl apply -f deployment.yml'
```

```
                    sh 'kubectl apply -f service.yml'
```

```
                }
```

```
            }
```

```
        }
```

```
}
```

After adding click on apply and then save it. Now click on build with parameters, choose apply as action click on build.

Instances | EC2 | my-cluster | Cluster | EC2 Instance Config | uber-role | IAM | EKS/terraform.tf | terraform-eks | pushandscanimage | Projects | a1a6ee537ec3d4... + - ⌂ X

Getting Started PRED National Portal of India NCERT Book List - APP... DCC Login Civil Engineering 5500... IES,MASTER_SSCE20... SSC 2016 - 19 GS QUE... Nadu Nedu Timetable - NCERTs.pdf APREIS - 2020 Home >

Dashboard > terraform-eks > Configuration

Configure

- General
- Advanced Project Options
- Pipeline**

```

1 * pipeline {
2     agent{ any }
3     stages {
4         stage('Checkout from Git'){
5             steps{
6                 checkout scm(branches: [[name: '*/master'], extensions: [], userRemoteConfigs: [[url: 'https://github.com/vommidapuchinni/ub']])
7             }
8         }
9         stage('Terraform version'){
10            steps{
11                sh 'terraform --version'
12            }
13        }
14        stage('Terraform init'){
15            steps{
16                dir('EKS_TERRAFORM') {
17                    sh 'terraform init'
18                }
19            }
20        }
21        stage('Terraform validate'){
22            steps{
23                dir('EKS_TERRAFORM') {
24                    sh 'terraform validate'
25                }
26            }
27        }
28        stage('Terraform plan'){
29            steps{
30                dir('EKS_TERRAFORM') {
31                    sh 'terraform plan'
32                }
33            }
34        }
35        stage('Terraform apply/destroy'){
36            steps{
37                dir('EKS_TERRAFORM') {
38                    sh 'terraform ${action} --auto-approve'
39                }
40            }
41        }
42        stage('Deploy to kubernetes'){
43            steps{
44                script{
45                    dir('k8s') {
46                        withKubeConfig(caCertificate: '', clusterName: '', credentialsId: 'k8s', namespace: '', restrictKubeConfig: true)
47                        sh 'kubectl apply -f deployment.yml'
48                        sh 'kubectl apply -f service.yml'
49                    }
50                }
51            }
52        }
53    }
54 }
55 }
```

Save Apply



Instances | EC2 | my-cluster | Cluster | EC2 Instance Config | uber-role | IAM | EKS/terraform.tf | terraform-eks | pushandscanimage | Projects | a1a6ee537ec3d4... + - ⌂ X

Getting Started PRED National Portal of India NCERT Book List - APP... DCC Login Civil Engineering 5500... IES,MASTER_SSCE20... SSC 2016 - 19 GS QUE... Nadu Nedu Timetable - NCERTs.pdf APREIS - 2020 Home >

Dashboard > terraform-eks > Jenkins

Status

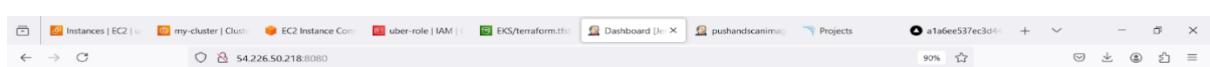
Changes Build with Parameters Configure Delete Pipeline Full Stage View Stages Rename Pipeline Syntax

Stage View

Average stage times: (Average full run time: ~1min 17s)

	Checkout from Git	Terraform version	Terraform init	Terraform validate	Terraform plan	Terraform apply/destroy	Deploy to kubernetes
Jul 19 16:22: commit	329ms	371ms	4s	3s	4s	4s	2s
Jul 19 16:18: commit	450ms	1s	6s	3s	4s	4s	1s failed
Jul 19 16:17: No Changes							
Jul 19 No	277ms	361ms	3s	3s	3s	2min 0s	

26°C Mostly cloudy



Instances | EC2 | my-cluster | Cluster | EC2 Instance Config | uber-role | IAM | EKS/terraform.tf | Dashboard | pushandscanimage | Projects | a1a6ee537ec3d4... + - ⌂ X

Getting Started PRED National Portal of India NCERT Book List - APP... DCC Login Civil Engineering 5500... IES,MASTER_SSCE20... SSC 2016 - 19 GS QUE... Nadu Nedu Timetable - NCERTs.pdf APREIS - 2020 Home >

Dashboard > Jenkins

Build History

+ New Item Build History Project Relationship Check File Fingerprint Manage Jenkins My Views

S	W	Name	Last Success	Last Failure	Last Duration
✓	cloud	pushandscanimage	2 min 53 sec #12	24 min #11	2 min 1 sec
✓	cloud	terraform-eks	7 min 20 sec #6	11 min #5	20 sec

Build Queue: No builds in the queue.

Build Executor Status: 1 Idle, 2 Idle

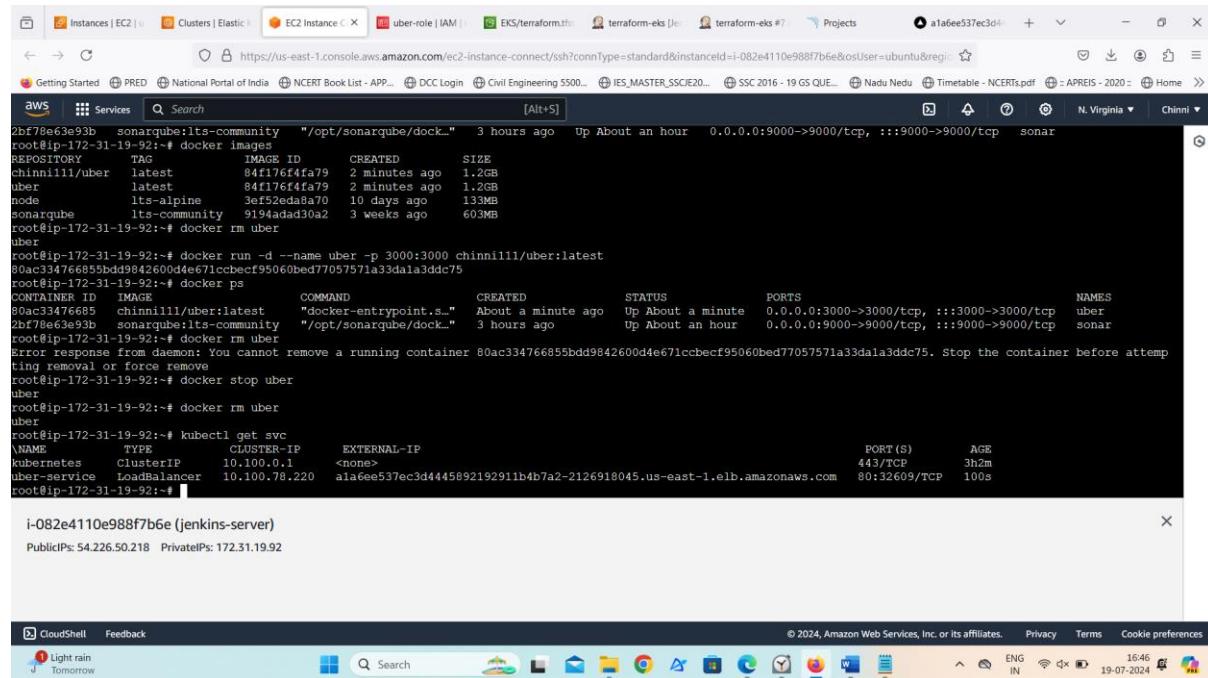
Add description



Step12: Accessing our application

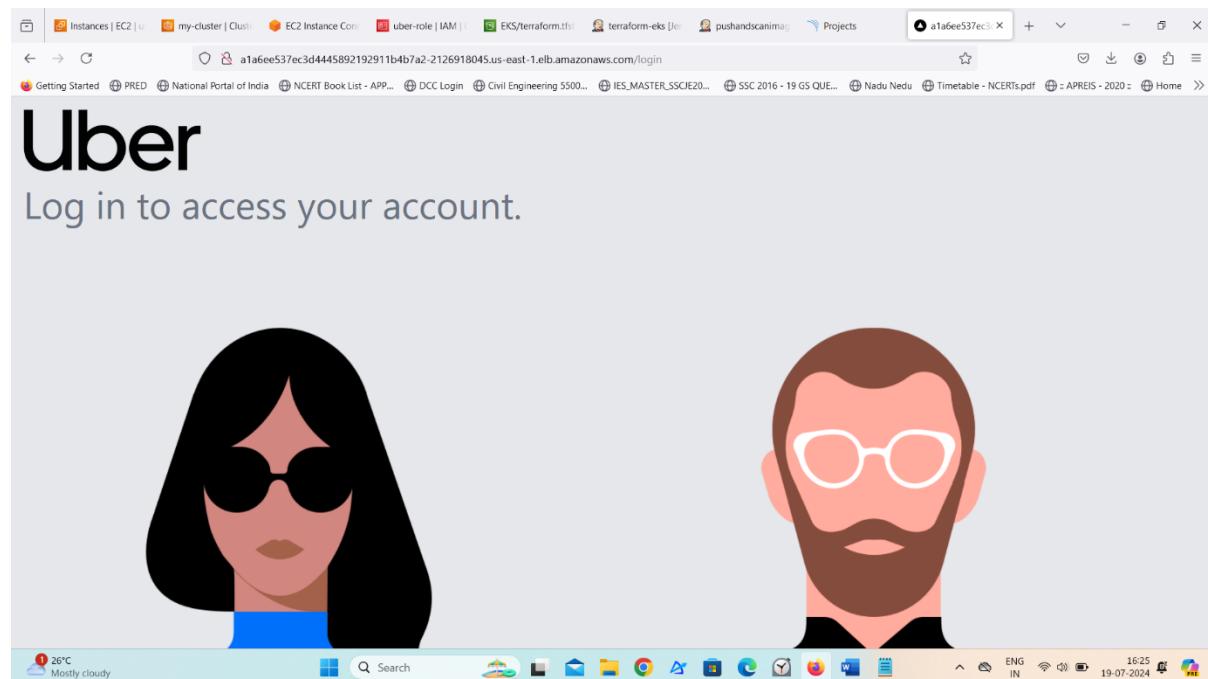
Go to server give command as kubectl get svc.

We see our services which are created.



```
2ba78e63e93b sonarqube:lts-community "/opt/sonarqube/docker..." 3 hours ago Up About an hour 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp sonar
root@ip-172-31-19-92:~# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
chinni111/uber     latest   84f176f4fa79  2 minutes ago  1.2GB
uber               latest   84f176f4fa79  2 minutes ago  1.2GB
node               lts-alpine 3ef52eda8a70  10 days ago   133MB
sonarqube          lts-community 9194adad30a2  3 weeks ago   609MB
root@ip-172-31-19-92:~# docker rm uber
uber
root@ip-172-31-19-92:~# docker run -d --name uber -p 3000:3000 chinni111/uber:latest
80ac334766855bdd9842600d4e671ccbe95060bed77057571a33dal1a3ddc75
root@ip-172-31-19-92:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS                               NAMES
80ac334766855bdd9842600d4e671ccbe95060bed77057571a33dal1a3ddc75. Stop the container before attempting removal or force remove
root@ip-172-31-19-92:~# docker stop uber
uber
root@ip-172-31-19-92:~# docker rm uber
uber
root@ip-172-31-19-92:~# kubectl get svc
NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes     ClusterIP  10.100.0.1    <none>          443/TCP         3h2m
uber-service   LoadBalancer 10.100.78.220  ala6ee537ec3d4445892192911b4b7a2-2126918045.us-east-1.elb.amazonaws.com  80:32609/TCP  100s
root@ip-172-31-19-92:~# i-082e4110e988f7b6e (jenkins-server)
PublicIP: 54.226.50.218 PrivateIP: 172.31.19.92
```

Copy load balancer external ip and paste on another tab we can see our webpage.



- Go to S3 bucket, click on created bucket.
- We see EKS folder in it. Click on it we see terraform statefile.

The screenshot shows the AWS S3 console with the path `Amazon S3 > Buckets > uberclone-bucket`. The 'Objects' tab is selected, displaying one item: `EKS/` (Folder). The folder was created by the user.

The screenshot shows the AWS S3 console with the path `Amazon S3 > Buckets > uberclone-bucket > EKS/`. The 'Objects' tab is selected, displaying one item: `terraform.tfstate` (tfstate). This file was uploaded by the user.

The screenshot shows the AWS S3 console with the path `Amazon S3 > Buckets > uberclone-bucket > EKS/ > terraform.tfstate`. The 'Properties' tab is selected, showing the following details for the `terraform.tfstate` file:

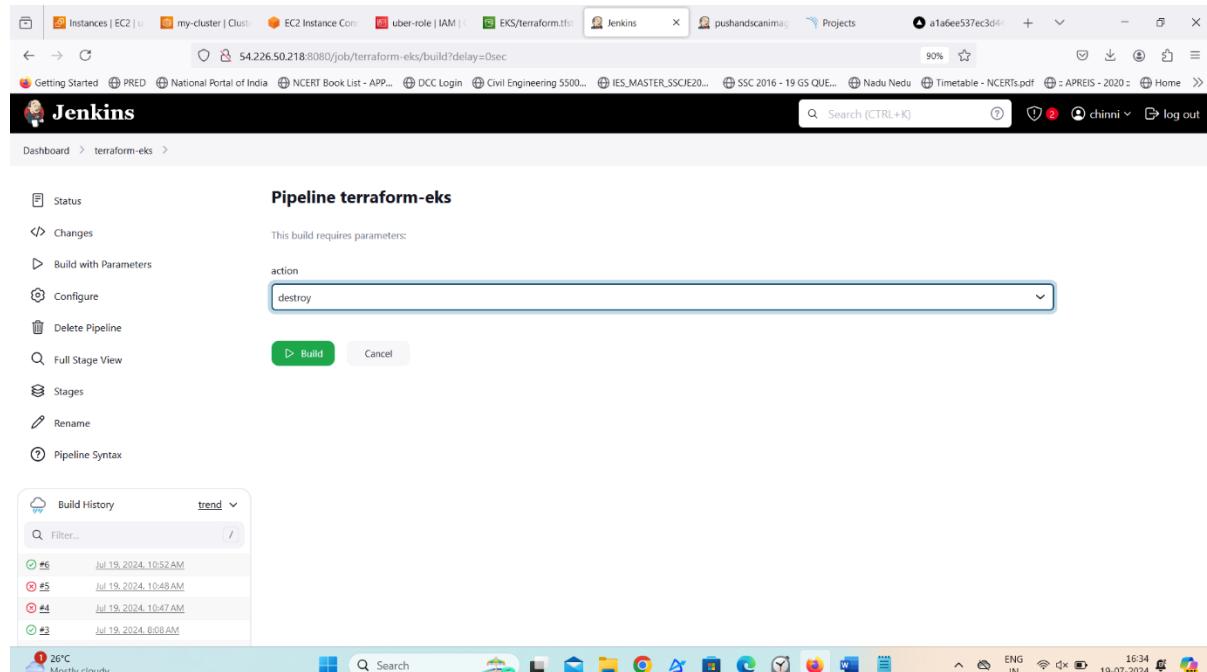
Key	Value
Owner	lokeshchinni111
AWS Region	US East (N. Virginia) us-east-1
Last modified	July 19, 2024, 13:41:12 (UTC+05:30)
Size	6.6 KB
Type	tfstate

Step13: Destruction

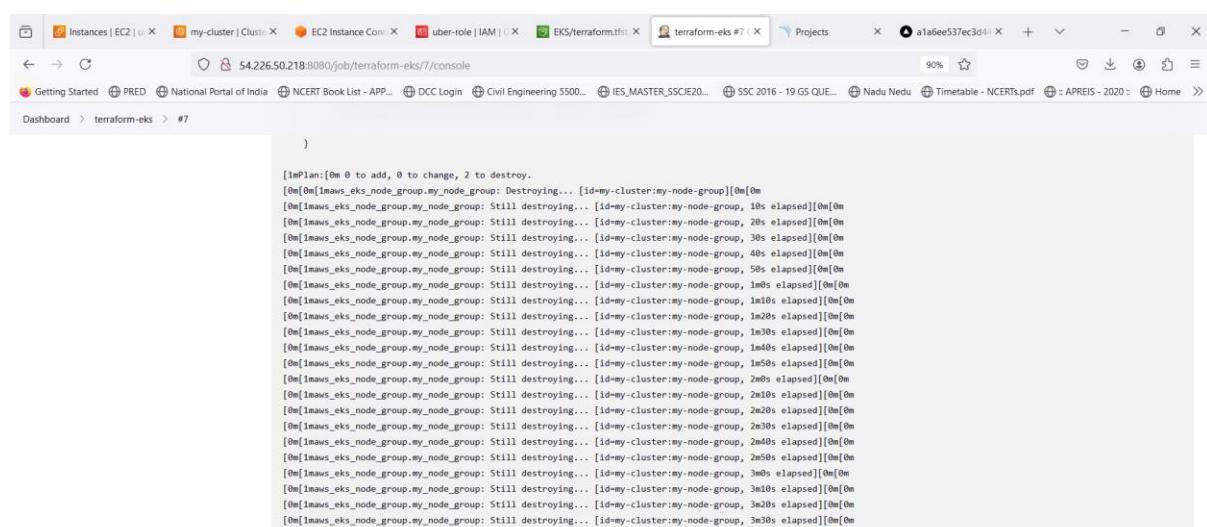
Go to terraform-eks pipeline. Click on build with parameters.

Choose destroy as action click on build.

Until it destroys don't delete anything.



The screenshot shows the Jenkins Pipeline configuration for 'Pipeline terraform-eks'. The 'action' dropdown is set to 'destroy'. The 'Build' button is highlighted in green. Below the pipeline configuration, the 'Build History' section shows five builds: #6, #5, #4, #3, and #2, all of which have a green checkmark indicating they were successful. The Jenkins interface includes a top navigation bar with various links and a status bar at the bottom showing the date and time.



The screenshot shows the Jenkins pipeline console output for build #7. The output displays the 'destroy' command being run repeatedly against the 'myaws_eks_node_group' node group. The log entries show the command being issued and the progress of the destruction process, with timestamps indicating the duration between each attempt. The Jenkins interface includes a top navigation bar with various links and a status bar at the bottom showing the date and time.



The screenshot shows the Jenkins pipeline console output for build #7. The output continues to show the 'destroy' command being run against the 'myaws_eks_node_group' node group. The log entries show the command being issued and the progress of the destruction process, with timestamps indicating the duration between each attempt. The Jenkins interface includes a top navigation bar with various links and a status bar at the bottom showing the date and time.

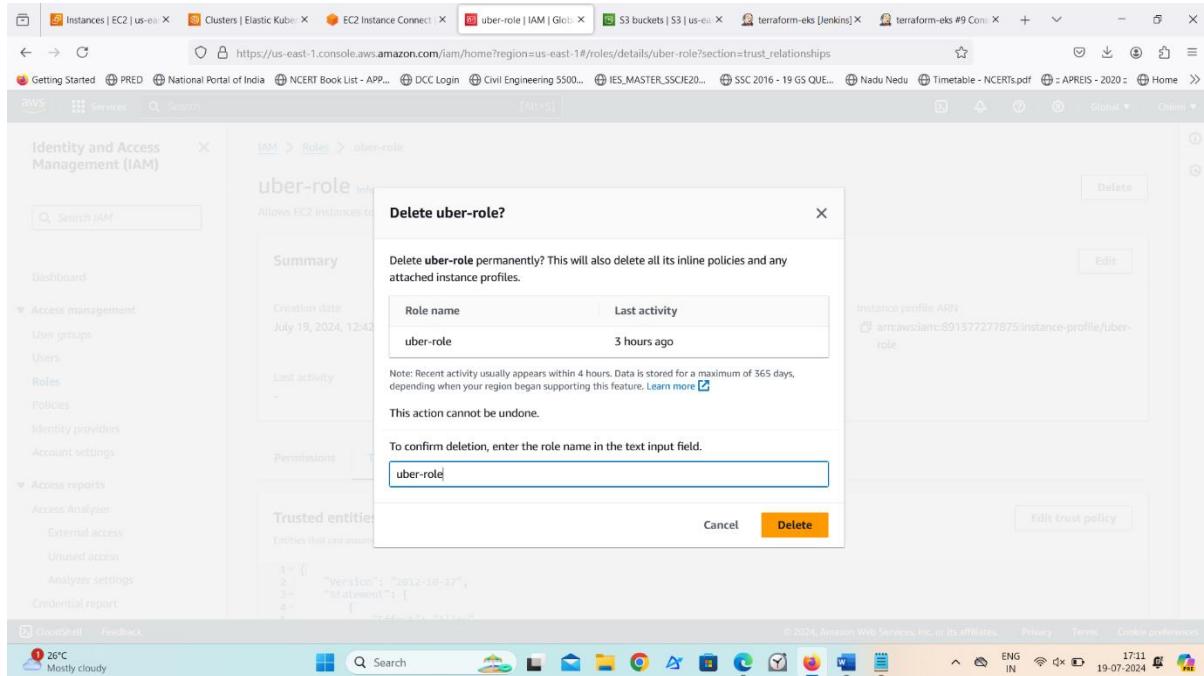
Delete two pipelines created and click on logout at right corner.

Logout from SonarQube server also.

Go to instance, click on instance state, click on terminate.

Go to S3 bucket and click on empty, after empty click on delete. Delete the bucket.

Also delete created IAM role.



Conclusion:

In conclusion, this documentation outlines a robust CI/CD pipeline tailored for our Uber clone project, aiming to optimize our development processes and streamline our delivery pipeline. By detailing each stage from code integration to automated testing and deployment, this documentation serves as a comprehensive guide for developers, testers, and operations teams. It emphasizes the importance of automation in ensuring code quality, enhancing collaboration, and achieving faster time-to-market. As we continue to iterate and improve our application, this CI/CD pipeline will play a pivotal role in maintaining reliability, scalability, and efficiency, ultimately contributing to the success of our Uber-like service.